Machine learning-based automated segmentation with a feedback loop for 3D synchrotron micro-CT

Zur Erlangung des akademischen Grades eines

Doktors der Ingenieurwissenschaften

von der KIT-Fakultät für Maschinenbau des

Karlsruher Instituts für Technologie (KIT)

genehmigte

Dissertation

von

M.Sc. Roman Shkarin

geboren am 11. Februar 1990 in Kargasok

Hauptreferent:	apl. Prof. Dr.-Ing. Ralf Mikut
Korreferenten:	Prof. Dr. Britta Nestler
	Prof. Dr. Tilo Baumbach

Tag der mündlichen Prüfung: 9. November 2020

Zusammenfassung

Die Entwicklung von Synchrotronlichtquellen der dritten Generation hat die Grundlage für die Untersuchung der 3D-Struktur opaker Proben mit einer Auflösung im Mikrometerbereich und höher geschaffen. Dies führte zur Entwicklung der Röntgen-Synchrotron-Mikro-Computertomographie, welche die Schaffung von Bildgebungseinrichtungen zur Untersuchung von Proben verschiedenster Art förderte, z.B. von Modellorganismen, um die Physiologie komplexer lebender Systeme besser zu verstehen. Die Entwicklung moderner Steuerungssysteme und Robotik ermöglichte die vollständige Automatisierung der Röntgenbildgebungsexperimente und die Kalibrierung der Parameter des Versuchsaufbaus während des Betriebs. Die Weiterentwicklung der digitalen Detektorsysteme führte zu Verbesserungen der Auflösung, des Dynamikbereichs, der Empfindlichkeit und anderer wesentlicher Eigenschaften. Diese Verbesserungen führten zu einer beträchtlichen Steigerung des Durchsatzes des Bildgebungsprozesses, aber auf der anderen Seite begannen die Experimente eine wesentlich größere Datenmenge von bis zu Dutzenden von Terabyte zu generieren, welche anschließend manuell verarbeitet wurden. Somit ebneten diese technischen Fortschritte den Weg für die Durchführung effizienterer Hochdurchsatzexperimente zur Untersuchung einer großen Anzahl von Proben, welche Datensätze von besserer Qualität produzierten. In der wissenschaftlichen Gemeinschaft besteht daher ein hoher Bedarf an einem effizienten, automatisierten Workflow für die Röntgendatenanalyse, welcher eine solche Datenlast bewältigen und wertvolle Erkenntnisse für die Fachexperten liefern kann. Die bestehenden Lösungen für einen solchen Workflow sind nicht direkt auf Hochdurchsatzexperimente anwendbar, da sie für Ad-hoc-Szenarien im Bereich der medizinischen Bildgebung entwickelt wurden. Daher sind sie nicht für Hochdurchsatzdatenströme optimiert und auch nicht in der Lage, die hierarchische Beschaffenheit von Proben zu nutzen.

Die wichtigsten Beiträge der vorliegenden Arbeit sind ein neuer automatisierter Analyse-Workflow, der für die effiziente Verarbeitung heterogener Röntgendatensätze hierarchischer Natur geeignet ist. Der entwickelte Workflow basiert auf verbesserten Methoden zur Datenvorverarbeitung, Registrierung, Lokalisierung und Segmentierung. Jede Phase eines Arbeitsablaufs, die eine Trainingsphase beinhaltet, kann automatisch feinabgestimmt werden, um die besten Hyperparameter für den spezifischen Datensatz zu finden. Für die Analyse von Faserstrukturen in Proben wurde eine neue, hochgradig parallelisierbare 3D-Orientierungsanalysemethode entwickelt, die auf einem neuartigen Konzept der emittierenden Strahlen basiert und eine präzisere morphologische Analyse

ermöglicht. Alle entwickelten Methoden wurden gründlich an synthetischen Datensätzen validiert, um ihre Anwendbarkeit unter verschiedenen Abbildungsbedingungen quantitativ zu bewerten. Es wurde gezeigt, dass der Workflow in der Lage ist, eine Reihe von Datensätzen ähnlicher Art zu verarbeiten. Darüber hinaus werden die effizienten CPU/GPU-Implementierungen des entwickelten Workflows und der Methoden vorgestellt und der Gemeinschaft als Module für die Sprache Python zur Verfügung gestellt.

Der entwickelte automatisierte Analyse-Workflow wurde erfolgreich für Mikro-CT-Datensätze angewandt, die in Hochdurchsatzröntgenexperimenten im Bereich der Entwicklungsbiologie und Materialwissenschaft gewonnen wurden. Insbesondere wurde dieser Arbeitsablauf für die Analyse der Medaka-Fisch-Datensätze angewandt, was eine automatisierte Segmentierung und anschließende morphologische Analyse von Gehirn, Leber, Kopfnephronen und Herz ermöglichte. Darüber hinaus wurde die entwickelte Methode der 3D-Orientierungsanalyse bei der morphologischen Analyse von Polymergerüst-Datensätzen eingesetzt, um einen Herstellungsprozess in Richtung wünschenswerter Eigenschaften zu lenken.

Abstract

The development of the third-generation synchrotron light sources has established the foundation for the investigation of the 3D structure of opaque samples at micrometer resolution and higher. That led to the development of X-ray synchrotron micro-computed tomography which promoted the creation of imaging setups aimed at the investigation of samples of diverse nature, for example, model organisms to better understand the physiology of complex living systems. The development of modern control systems and robotics allowed for the complete automation of the X-ray imaging experiments and calibrating the experimental setup parameters on the fly. The advancement of digital detector systems prompted enhancements in resolution, dynamic range, sensitivity, and other essential characteristics. These improvements resulted in a substantial increase in the throughput of the imaging process, but on the other side, experiments started to produce a substantially increased amount of data up to dozens of terabytes that were processed manually afterward. Thus, these technical advancements paved the way for carrying out more efficient high-throughput experiments to study a large number of samples producing datasets of better quality. Thereby, in a scientific community is a high demand for an efficient automated X-ray data analysis workflow that could cope with such data burden and provide valuable insights for the domain experts. The existing solutions for such a workflow are not directly applicable to high-throughput experiments since they were developed for ad-hoc scenarios in the medical imaging domain. Therefore, they are not optimized for high-throughput data streams and incapable to exploit the hierarchical nature of samples.

The major contributions of the present thesis are a new automated analysis workflow suited for efficient processing of heterogeneous X-ray datasets having hierarchical nature. The developed workflow is based on improved cutting-edge methods aimed at data pre-processing, registration, localization, and segmentation. Every stage of a workflow involving a training phase can be automatically fine-tuned to find the best hyper-parameters for the specific dataset. To address the analysis of fibrous structures in samples, a new highly parallelizable 3D orientation analysis method was developed based on a novel concept of emitting rays, which allowed for more precise morphological analysis. All the developed methods were thoroughly validated on synthetic datasets to quantitatively assess their applicability in different imaging conditions. It was shown that the workflow is able to process a series of datasets of similar nature. Moreover, the efficient CPU/GPU implementations of the developed workflow and methods

were presented and made available for the community as modules for Python language.

The developed automated analysis workflow was successfully applied for micro-CT datasets acquired in high-throughput X-ray experiments in the field of developmental biology and material science. In particular, this workflow was applied for the analysis of the medaka fish datasets, which allowed for automated segmentation and subsequent morphological analysis of the brain, liver, head nephrons, and heart. Moreover, the developed 3D orientation analysis method was employed in the morphological analysis of polymer scaffolds datasets to guide a fabrication process towards desirable properties.

Acknowledgements

Foremost, I would like to express my gratitude to my supervisor Prof. Dr. Ralf Mikut for his scientific guidance and ongoing interest in the progress of this work, without his leadership and dedication, this thesis would not be possible. It has been a privilege working with you and getting to know you.

I am very grateful to Prof. Dr. Tilo Baumbach for his leadership, providing excellent working conditions, fruitful discussions, and being my second supervisor for this work.

I want to warmly thank Dr. Venera Weinhardt for her many comprehensive advice and frankness towards my steps in a scientific career (which ultimately led to writing this thesis). Without her supportive guidance, I would not correctly carry out any experiment and write any papers.

I thank Prof. Dr. Joachim Wittbrodt, Dr. Felix Loosli, and Naren Aadepu for their support in the preparation of experiments with medaka fish organisms and genuine interest in my work.

Furthermore, I would like to thank all colleagues for their warm welcome, contributions to the working atmosphere, and inspiring scientific discussions. I very much appreciate the scientific and personal support I have received from each of you individually, in particular: Sabine Bremer, Marcus Zuber, Janes Odar, Alexey Ershov, Tomaš Farago, and Thomas van de Kamp.

Moreover, I want to thank the X-Regio project supported by the Bundesministerium für Bildung und Forschung (BMBF) and Karlsruhe Haus of Young Scientists (KHYS) for financial support of my doctoral research phase.

I acknowledge Dr. Roman Surmenev and Dr. Maria Surmeneva for providing polymer and composite scaffolds.

I also would like to thank institute administration, especially Ekaterina Walter, Margit Helma, Esra Aran, and Sandra Richers, who were for me whenever I needed their help to solve bureaucratic problems.

Last but not least, I would like to express my deep appreciation to my dear family, without whom I could not bring this work to the end. I am infinitely grateful to my mum who gave me love and support to start my academic career path. To my brother Andrei, who supported and helped me in doubtful moments in my work and life. To my beloved wife Svetlana who is not only my life partner but also a reliable and careful colleague, who always motivated and supported me in

my scientific career from day one, without her care and love this path would be impossible for me.

Contents

Contents

1. Introduction

1.1. Motivation

Man always was curious about the world of tiny insects and things, whose details were hidden from the naked eye. Soon, these perception limits were overcome by discovering lenses during the Roman period. The people were experimenting with various shapes of glass, and it was observed that the thick in the middle and thin on the edges glass could magnify an object. However, they were forgotten and not used much for centuries, and were recalled only at the end of the 13th century by spectacle makers who were producing glasses made of a pair of lenses. It is the time when the earliest microscopes emerged being just magnifying glasses with a constant power of 6-10x. It was called as "flea glasses" because, at that time, fleas and other small insects were favorite subjects for observations. At the end of the 16th century, two spectacle makers, Zaccharias Janssen and his father Hans during experimenting with lenses made a significant discovery [1]. They placed several lenses in a tube and observed that such combination of lenses provides much higher magnification than a single lens. Although, their microscope can hardly be called a scientific tool because the magnification power was still too small, around 9x and images were somewhat blurry. The world's first practical microscope was invented by Antony Van Leeuwenhoek, a draper, and scientist. His microscope was hand-held and had a single lens [2]. The secret was a superior lens which he made by grinding and polishing a small glass ball, providing magnification of 270x, the finest known at that time. This microscope allowed people to see things that no one had ever seen before like yeast, blood cells, bacteria and many tiny animals swimming in a drop of water. Until that time people did not realize that magnification can reveal previously hidden structures, and change minds about the complexity of the world, that all life might be made up of tiny pieces unseen by the naked eye. In the middle of the 17th century, Robert Hooke had improved the microscope by use of a three-lens configuration, which allowed him to discover a basic unit of life, a cell. All early microscopes provide quite distorted images due to glass quality and imperfections of lenses [2]. During the 19th century, vast improvements in quality had been made by Zeiss and Charles Spencer companies producing fine optical instruments [3]. However, soon the development of microscopes was slowed down since physical principles of optics were well understood, and the limits of optics were reached. After that, the majority of produced microscopes follow similar construction principles which inherent in monocular or binocular microscopes. The optical microscope

was mainly used for imaging sectioned or transparent samples. However, soon a discovery will permit investigations of opaque samples.

In the last years of the 19th century, Wilhelm Röntgen discovered a phenomenon which will change the world and effect on all fields of life and science [4]. The discovery of X-rays called in honorous of Röntgen, allowed for non-invasive investigations of living organisms and non-destructive control of materials. Within a few years, these rays became a traditional medical tool for a more efficient treatment of broken bones, discovering pathologies, diagnostic and therapeutic purposes in various medical disciplines. In the next year after discovery, Thomas Eddison developed the fluoroscope which became the standard for medical X-ray experimentation. In the same year, Hall-Edwards performed a surgical operation with the help of X-rays. A decade after the discovery, the physicist Charles Barkla found that gases could scatter X-rays and that each element has a characteristic X-ray spectrum. The unique properties of this radiation were proactively applied in numerous areas of human life, for instance, as an assisting tool for the fitting of shoes in retail shoe stores. In the next decade, the initial step towards cross-sectional imaging has been done by Johann Radon, who established the mathematical basis for the reconstruction of a three-dimensional (3D) object from an infinite number of two-dimensional (2D) projections of this object. Later, these principles were applied by Godfrey Hounsfield and Allan Cormack in their the world's first computed tomography (CT) scanner [5]. Further advances in physics will stimulate the birth of other imaging modalities as Magnetic resonance imaging (MRI), Positron-emission tomography (PET), Single-photon emission computed tomography (SPECT), and others, which successfully will have found their place in the medical domain.

A couple of decades after the discovery of X-rays, physicists have gained interest in the radiation as an energy-loss mechanism, and magnetic induction electron accelerators producing intense beams of X-rays have been built. The synchrotron radiation was accidentally discovered in the middle of the 20th century at General Electric (GE) cyclotron because the machine was not entirely shielded and the coating of the doughnut-shaped electron tube was transparent that allowed a technician unintentionally see sparking in the tube with a large mirror. The research group quickly understood his observation of a bright arc of light coming from the electron beam [4]. Afterward, some improvements were introduced to cope with the disadvantages of cyclotrons for imaging purposes, which resulted in the modern third-generation synchrotron light sources providing the characteristics of the beam greatly superior to laboratory X-ray sources.

The discovered X-ray source provided a significantly higher brilliance and photon flux that allowed for much shorter acquisition time to acquire the data of a comparable quality to conventional X-ray sources that have a low number of incoming photons, poor focusing, and non-parallel geometry. The third-generation synchrotron light sources have established the basis for investigation of the 3D structure of opaque samples at micrometer resolution and higher. That led to the emergence of X-ray synchrotron micro-computed tomography (SR-μCT) which

promoted the creation of tailored imaging setups aimed at investigation of samples of different nature.

It was widely applied for revealing of 3D structures of biological samples from many species of organisms including human, mouse, fish, worm, and insect. In material science, it allowed for the identification of internal material inhomogeneities, assessment of internal defects and damage, and investigation of microstructural-property relationships.

Evolution of digital electronics allowed for rapid development and improvement of digital detectors in resolution, dynamic range, sensitivity, and others characteristics, which are exceptionally important for any imaging setup. The modern detectors have a significantly higher number of pixels than predecessors, allowed to acquire more information on radiograms, a better resolution to capture finer details, improved sensitivity of pixels and dynamic range enabling to detect even subtle variations in absorption of a specimen matter. However, another side of such improvements is a significantly increased size of each projection image.

The emergence of modern control systems and advances in robotics allowed for automation of sample change, systems of servo motors to adaptively control a position of a sample and all components involved in the experiment [6]. Thus, series of samples can be studied without visiting the experimental room, which is essential for investigation of specimens which are sensitive to even a subtle change of environmental conditions, evaluation of the phenotype-genotype association, and drug action on morphometry of internal structures of living organisms. These advances allow managing the whole imaging process automatically and adjusting the experimental parameters on the fly, which leads to dramatically increase of the throughput of imaging setup and, as a consequence of the amount of produced data which must be processed after the experiment.

To obtain valuable information from the acquired data, it should be processed and analyzed first by the experts in the field such as biologists, doctors, material scientists, physicists, and others. The analysis of such data usually consists of a delineation of sample structures to make the further morphological analysis of structures and density of matter possible. The delineation process is called segmentation, and it allows partitioning data into distinct sets of pixels or voxels, so-called labels, describing shapes of target structures. Such a representation facilitates quantitative analysis of structures, which consists of finding the volume, surface, density, etc. However, segmentation is one of the hardest tasks in the image analysis field, since it requires precise delineation of structures at each slice of data. For example, an experienced biologist could easily spend days and weeks on segmentation of data representing a highly detailed part of a sample [7].

Thus, experiments involving a few samples usually do not demand complete automation of data analysis since sometimes treat them manually or semi-automatically is quicker. However, the data produced in the result of high-throughput scanning experiments are impossible to be processed manually, especially in a phenotype-genotype association, and drug action on the morphom-

etry of living organisms, where the number of samples exceeds hundreds. That creates an urgent need for efficient automated data analysis workflows able to handle such an immense amount of data.

1.2. X-ray computed tomography

1.2.1. History of synchrotron radiation

The synchrotron radiation generated by relativistic electrons in circular accelerators was born a few years after the Second World War. However, the beginning of synchrotron radiation started from the discovery of the electron in 1897 by John Thomson, which laid the theoretical basis for further development. In the same year, the French physicist Joseph Larmor derived an expression from classical electrodynamics for the total power radiated by an accelerated charged particle. Soon after, Alfred Liénard demonstrated that electrons moving along a circular path emit the power proportional to $(E/mc^2)^4/R^2$, where E is the kinetic energy of electrons, m is the electron rest mass, and R is the trajectory radius.
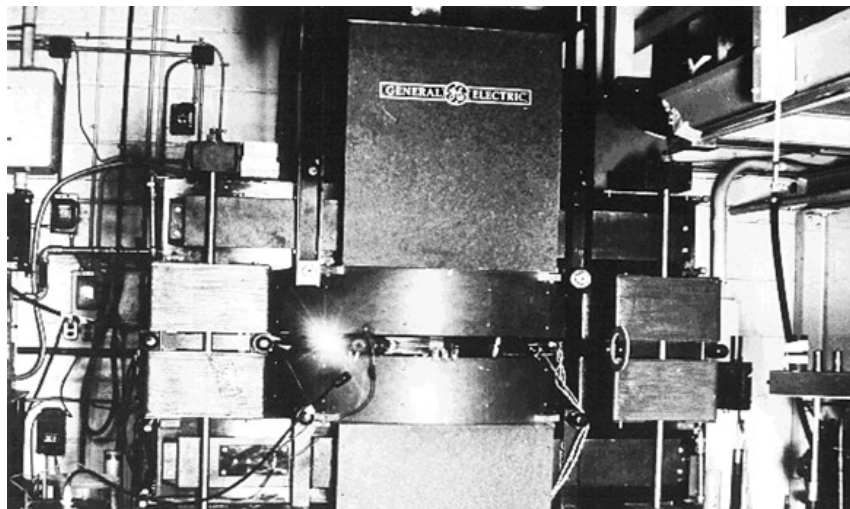
Figure 1.1.: The first-ever observation of synchrotron light from the 70 MeV electron synchrotron at General Electric Research Laboratory. Reproduced from [8].

The first time synchrotron radiation was detected at cyclotron (Fig. 1.1) of the General Electric Research Laboratory in Schenectady (New York) on 24 April 1947. Since this was the first time, the phenomenon was not immediately recognized and understood. Here is the fragment of a letter dated 25 September 1970 from Herb C. Pollock to Dmitri Ivanenko describing this event after almost 20 years later:

> *On April 24, Langmuir and I were running the machine and as usual were trying to push the electron gun and its associated pulse transformer to the*

limit. Some intermittent sparking had occurred and we asked the technician to observe with a mirror around the protective concrete wall. He immediately signaled to turn off the synchrotron as 'he saw an arc in the tube.' The vacuum was still excellent, so Langmuir and I came to the end of the wall and observed. At first we thought it might be due to Cerenkov radiation, but it soon became clearer that we were seeing Ivanenko and Pomeranchuk radiation.

The observed phenomenon of synchrotron radiation was predicted by two Russian physicists D. Ivanenko and I. Pomeranchuk in 1944 [9], and several years later was experimentally confirmed and formally described [10]. In the beginning, the synchrotron radiation was treated as an unwanted but unavoidable loss of energy in magnetic induction electron accelerators designed to produce intense beams of X-rays by directing an accelerated beam to an appropriate target. In 1956, Diran Tomboulian and Paul Hartman were granted for two-weeks access to the 300 MeV Cornell Synchrotron to conduct some experiments to investigate opportunities for further application of synchrotron light. Despite the limited time, they reported on potential advantages of using such X-ray sources, presented the experimental setup and photographic-plate in their paper [11], demonstrating the broadband synchrotron radiation emitted by monoenergetic electrons [11].

However, despite the advantages of synchrotron light source which the Cornell scientists highlighted in their work, only five years later an experimental program exploiting synchrotron radiation began. The National Bureau of Standards (now National Institute of Standards and Technology) modified its 180-MeV synchrotron in Washington, D. C. to allow access to the radiation through a tangent section into the machine's vacuum system. It was a birth of Synchrotron Ultraviolet Radiation Facility (SURF I), the first facility providing access to X-ray source for external users.

The first generation of synchrotron radiation sources was usually referred to as parasitic facilities, mainly due to the primary users of these facilities of that time, since the synchrotrons initially were designed for high-energy or nuclear physics experiments. Along with SURF I, facilities in Frascati and Japan soon after launch began to attract regular physicists eager to explore the capabilities of synchrotron radiation. In the beginning, experiments were conducted in the ultraviolet and soft X-ray modes, because most of these early facilities had storage-ring energies around 1 GeV. In 1964, the Deutsches Elektronen-Synchrotron (DESY) with the storage-ring of 6 GeV in Hamburg opened the doors for the first users, providing the extended energy spectrum up to the hard x-ray regions down to 0.1 Å(about 125 keV). Many other facilities started to attract synchrotron radiation users, one of the first was the 240 MeV Tantalus synchrotron in Wisconsin. Even though it was not originally designed to supply synchrotron radiation, it became the first facility exclusively providing the beam for synchrotron experiments.

The advances in the engineering of efficient electron storage rings optimized for the long-term operation were a catalyst for the development of the first dedi-

cated facilities aimed exclusively at synchrotron radiation. The first such facility was the 2 GeV Synchrotron Radiation Source (SRS) at Daresbury, England and experiments began in 1981. Due to introduced improvements in design, these facilities started to call as "second-generation" synchrotron light sources. As time passed, many new facilities were built and commissioned, while some of the first-generation synchrotrons were upgraded to second-generation.

After the invention of the second-generation facilities, it became clear that enhancement of X-ray beam brilliance could be achieved by optimizing of "emittance" property of the electron beam with the specialized devices as the array of magnets (the "magnet lattice") intended to manipulate the beam, and so-called "insertion devices". Among them are "wigglers" and "undulators", they are situated in the straight sections in between the curved arcs of storage rings. They function by perturbing the path of the electrons in an oscillatory manner, producing synchrotron radiation without effect on their average direction. The integration of insertion devices into storage rings marked the beginning of the third-generation of synchrotron sources, which were designed to provide optimal brilliance. The first implementation of the specification for third-generation facilities was done at the 6 GeV European Synchrotron Radiation Facility (ESRF) in Grenoble, France, which began operating in 1994.

At the moment, the most recent third-generation facilities are the National Synchrotron Light Source II (NSLS II) at the Brookhaven National Laboratories in Upton, New York, and MAX IV in Lund, Sweden. Construction of these facilities was started almost at the same time, NSLS II in 2009 and MAX IV in 2010. The facilities enable the study of material properties and functions at nanoscale resolution, providing world-leading capabilities for X-ray imaging. These significant improvements were made possible by recent advances in the construction of magnets for storage rings.

The development of the fourth-generation of synchrotron facilities has already begun. These facilities will be defined by a vastly enhanced performance, especially concerning the coherence and brilliance of the X-rays, using so-called energy recovery linacs (ERLs), and free electron lasers (FELs). With the help of them, the peak brilliance can be improved a further three and ten orders of magnitude using ERLs and FELs respectively. At the moment, the Linac Coherent Light Source (LCLS) at Stanford, and the FLASH and European XFEL facilities at Hamburg are presently on the way to this goal.

1.2.2. Synchrotron construction

The third-generation synchrotron light sources are characterized by their highly parallel and narrow X-ray beams of high intensity, moreover, all their parameters can be expressed via a single "brilliance" property. Another essential characteristic is the energy of the electrons moving within the storage ring and generating

the synchrotron radiation, in its turn, it influences on the spectrum of photon energies that facility can cover. The schematic depiction of the main components of a synchrotron is presented in Fig. 1.2.
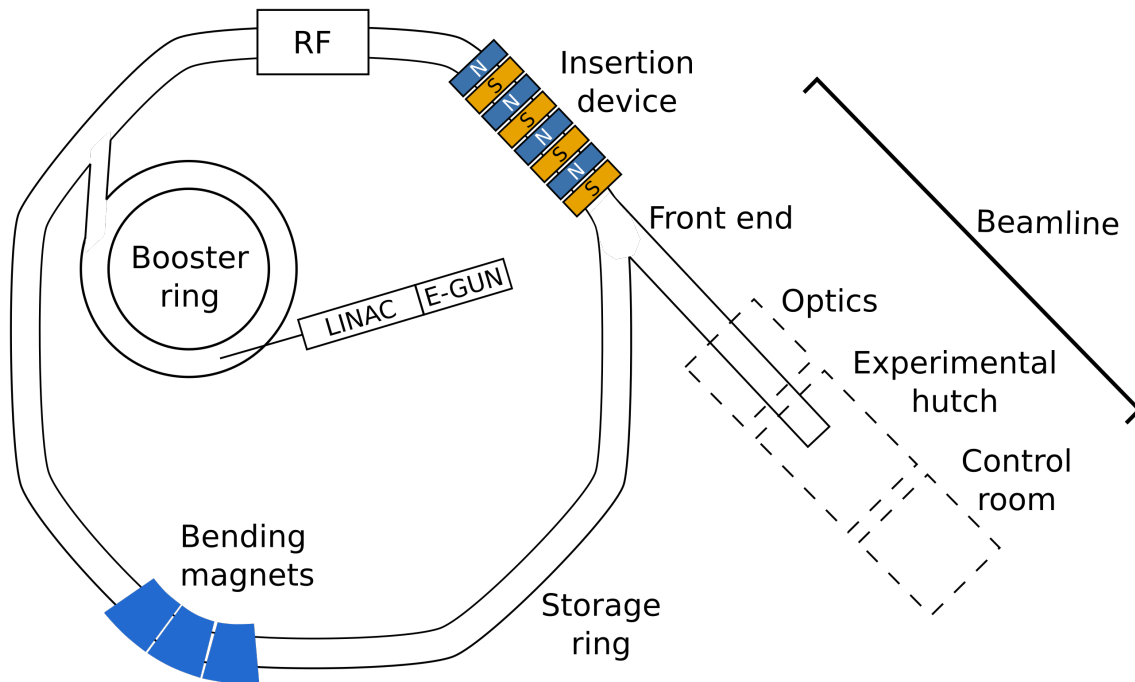
Figure 1.2.: The schema of the essential components of any modern synchrotron light source. Electrons produced by a heated filament in an electron gun are accelerated in a linear accelerator (linac) and transmitter into a booster ring, where they continue acceleration. After that, they are injected into the so-called storage ring, where they are circulating in a closed path with the help of bending magnets at arc sections. The beamlines use the radiation emitted from the bending magnets and insertion devices (wigglers and undulators). The energy loss of electron due to the emission of synchrotron radiation is replenished by a radio frequency (RF) supply.

1. The source of electrons is an electron gun, where thermionic emission from a hot filament usually generates the electrons. Then, they are accelerated with a linear accelerator (linac) to approximately 100 MeV. Since electrons collide with residual gas particles in the storage ring and regularly get lost, it is necessary to produce them continuously.

2. The electrons from the linac are injected into a booster ring where they are accelerated. The target energy for acceleration might either be equal to the energy in the main storage ring or to some lower energy. When the target energy is reached, they are periodically injected into the storage ring to maintain the specified storage ring current.

3. Electrons circulate inside the storage ring where they are being maintained on a closed path by an array of magnets, which is usually referred to as the "magnet lattice" of the ring. The most commonly used magnets are dipole- or bending magnets which induce the electrons to change their path

and follow the closed path, quadrupole magnets focus the electron beam and compensate for Coulomb repulsion between the electrons, and sextupole magnets correct chromatic aberrations caused by the focusing by the quadrupoles (Fig. 1.3).

Figure 1.3.: The components of the magnet lattice used at the KIT synchrotron light source. (a) Bending- or dipole magnets which force electrons to travel along a curved path, thereby emitting synchrotron radiation. (b) A focusing quadrupole magnet and (c) a correcting sextupole magnet.

The storage ring comprises arced sections containing bending magnets (BMs), and straight sections which host insertion devices (IDs) providing the most intense synchrotron radiation. The BMs deviate the electrons so that they move along the arced sections that connect the straight parts which are also used to provide bending-magnet radiation, although their brilliance is significantly lower than that produced by IDs.

4. Due to the emission of synchrotron radiation, the electrons lose energy. To replenish this energy, a radio frequency (RF) supply is used, which supplies the electrons with the extra amount of energy every time they pass through it.

5. The beamlines are placed along the axes of the insertion devices, which are tangentially connected to the bending magnets of the storage ring. When the beam comes to the beamline, at first it goes through the "front end", which isolates the beamline vacuum from the storage ring vacuum, it controls the position of the beam, blocks the X-ray and Bremsstrahlung radiation. The next section of the beamline contains the optics and experimental hutches, and it allows to filter out the low-energy tail of the synchrotron radiation spectrum, which can easily damage the optical components due to strong absorption. Then, the beam is focused and monochromated (if required), before it enters the experimental hutch. To protect the users not only from X-rays but also from gamma rays and high-energy neutrons, the hutch is usually shielded with lead-lined, thick concrete. Therefore, the experiments are performed remotely from the radiation-free area.

1.2.3. Advantages of synchrotron-based X-ray setups over laboratory CT systems

The laboratory computed tomography (CT) systems are becoming more afford-able for private companies, and research institutes than the synchrotron light sources, which are only a few in a country and perspective users require to pass special procedures to get the access. In comparison to the synchrotron X-ray facilities, they are typically operating with the cone beam geometry using the raw spectrum of the X-ray source, whereas synchrotron-based CT setups can do measurements using monochromatic X-rays with a parallel beam geometry (Fig. 1.4).
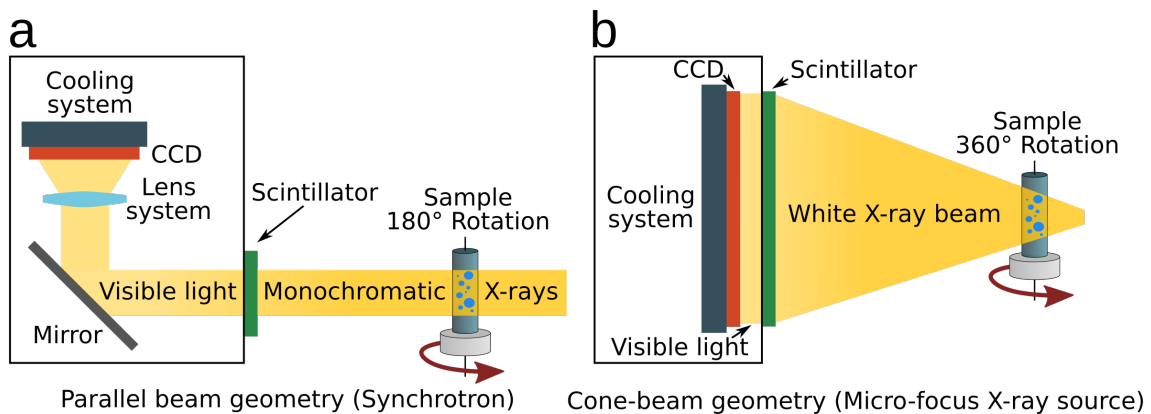
Figure 1.4.: The projection geometries of (a) synchrotron-based X-ray setup and (b) laboratory CT system, the latter produces a cone beam and enables to adjust the magnification and spatial resolution of the sample by changing its position on the X-ray optical path. Whereas, in the synchrotron-based X-ray setups, the magnification can be only achieved by modification of the optical system, and the distance between the sample and the scintillator influences on the severity of phase-contrast effects.

The synchrotron light sources provide three the most significant advantages over the laboratory CT systems, namely, the wide energy range (4–200 keV), high flux and small source size and beam divergence. Due to these properties, thick and dense samples can be examined in a shorter period. Moreover, the features of spatial coherence make possible to use phase-contrast techniques for imaging of weakly absorbing materials. The high photon flux enables to do monochromatic beam imaging with very narrow energy bandwidths ($\Delta E/E \sim 10^2 - 10^4$), that is helpful for quantitative absorption and phase imaging, and for imaging close to absorption edges of specific chemical elements. The synchrotron-based CT setups provide 10–100 times faster data acquisition for comparable spatial resolutions. The laboratory CT systems are still struggling to go below of 5 μm spatial resolution. Therefore, for the analysis of the soft-tissue morphology of millimeter-sized living samples, where phase-contrast and submicron-level spatial resolution are required, the synchrotron light sources are the only solution.

1.3. Experiment design

1.3.1. Synchrotron-µCT imaging process

The experimental setup, which is usually implemented at beamlines for the acquisition of synchrotron-µCT (SR-µCT) data is depicted in Fig. 1.5. The sample rotates 180/360° with the constant speed around the axis perpendicular to the incident X-ray beam. The signal produced by attenuation of the beam transmitted through the sample is recorded at regular intervals during the rotation with the help of a scintillator, an optics setup, and a digital detector.
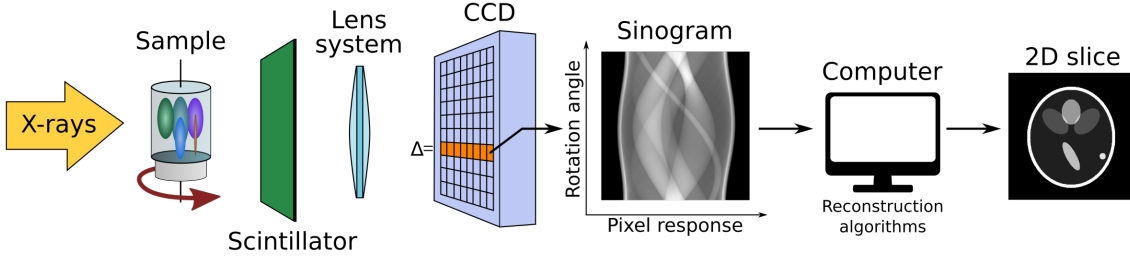
Figure 1.5.: The principle of X-ray computed tomography. A parallel beam of X-rays passes through a sample and falls on a scintillator to transform X-rays into visible light, which can be optionally magnified with a lens system onto a CCD detector. The responses from each row of pixels are collected at each rotation step into a stack to form the so-called sinogram, which is then fed into a computer. Then, a cross-section is reconstructed from a given sinogram at the corresponding height level using tomographic reconstruction algorithms. Finally, the 3D reconstruction is created from a stack of cross-sections.

The geometric resolution of the acquired data depends on several factors, namely the point-spread function of the scintillator, the magnification factor of the optics setup M, the signal-to-noise ratio of the detector, and the linear pixel size Δ and the point-spread function of the detector. The minimal number of projections N_{mp} which must be recorded is determined with the help of the Nyquist-Shannon sampling theorem and is equal to

$$N_{\mathrm{mp}} = \frac{N_{\mathrm{px}}\pi}{2}, \tag{1.1}$$

where N_{px} is the number of pixels in the row. The reconstructed object cross-section also known as a slice is produced from each row of a detector. The signal response of the N_{px} pixels from each row is collected at each rotation position, which can be presented as a function of rotation angle from the response, known as a sinogram (Fig. 1.6). The response of a detector pixel is proportional to the intensity of the transmitted X-rays that hit it after passing through the object along a path $L_{\mathrm{ray}} = M\Delta_z$, that is

$$I = I_0\, e^{-\mu_1 \Delta_z}\, e^{-\mu_2 \Delta_z}\, e^{-\mu_3 \Delta_z}\, \ldots\, e^{-\mu_M \Delta_z} \tag{1.2}$$

$$I = I_0 \exp \sum_{k=1}^{M} \mu_k \Delta_z. \tag{1.3}$$

The projection p for a given pixel at a given angle is determined as:

$$p = -\ln \frac{I}{I_0} = \sum_{k=1}^{M} \mu_k \Delta_z \approx \int_{L_{\text{ray}}} \mu_z dz. \tag{1.4}$$
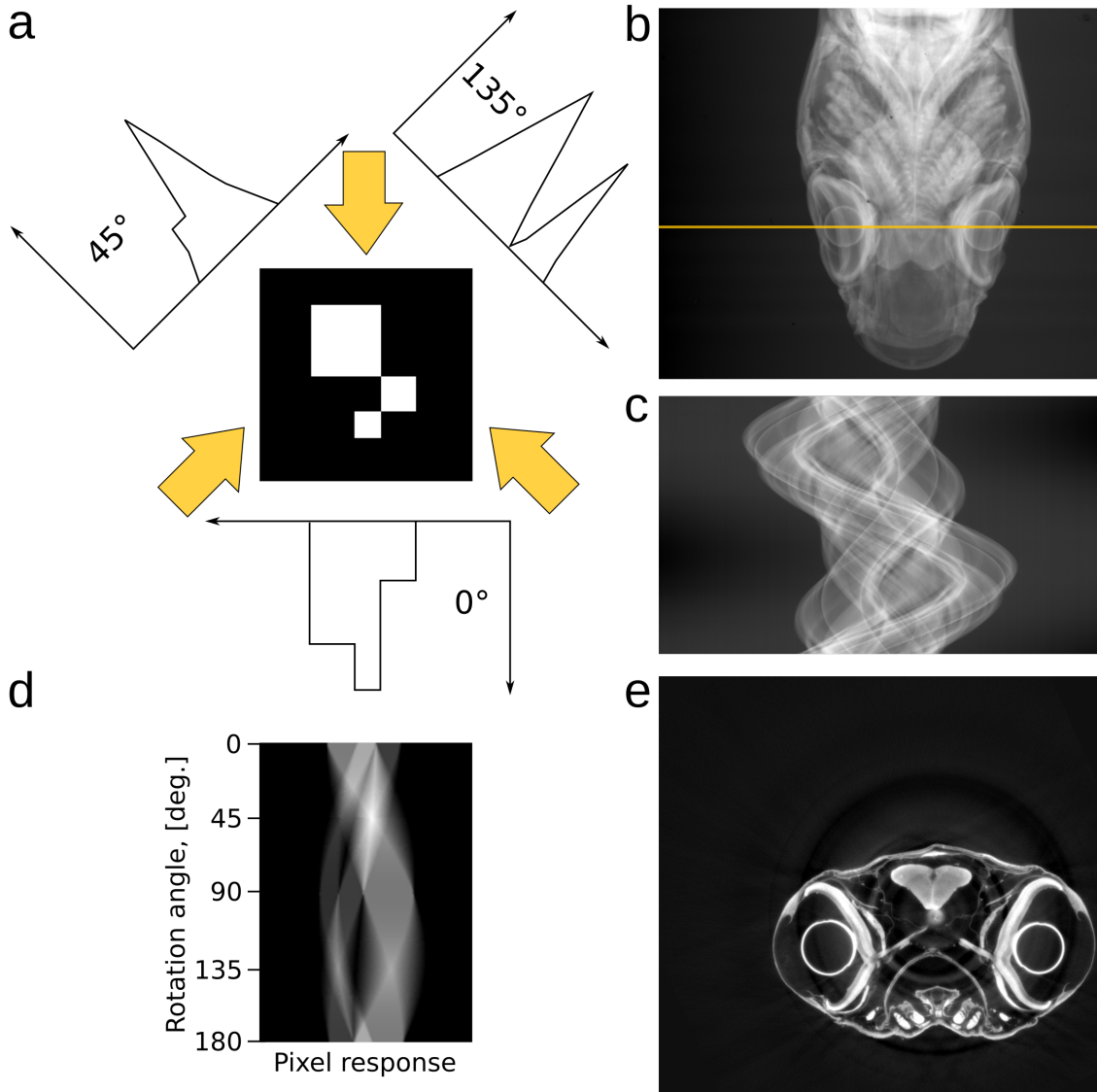
Figure 1.6.: The process of generating sinogram. An example of (d) a simple sinogram generated in the result of rotation around 180° (a) three absorbing blocks touching each other by corners. The representative transmission profiles at three symmetrical angles of 0°, 45°, 135°, which corresponds to 0th, 45th and 135th rows of the sinogram in a case of 1° rotation step. One of the tomograms acquired from (b) the top part of medaka fish, (c) the sinogram created by collecting rows of pixels from each tomogram marked by an orange line, and (e) the reconstructed slice from the corresponding sinogram.

1. Introduction

To reconstruct the attenuation distribution of the entire object, the line integrals for different angles for every pixel must be collected. Every slice is reconstructed from its respective sinogram using the preferable reconstruction algorithm. The most popular method is filtered back projection (FBP) [12], which is based on the Radon transform proposed by Johann Radon in 1917 [5].

The basic idea of FBP is presented in Fig. 1.7, there is a centered 2D absorbing disc. At the first step, the absorption profile of X-ray is collected in one direction. Then, this projection is run back through the image, whereby the projection intensity is evenly distributed among all the pixels along each ray path. It is done for a full set of projection angles spanning 180°. As a result, the projection of each respective angle is back projected and added to the image. The reconstruction quality can be improved by decreasing the angular step, which increases the number of acquired projections, and subsequently, the resemblance of the result of overlapping to the original object (Fig. 1.7). The plain back projection does not work well because it causes significant distortions in the reconstruction due to overlapping projections producing starlike artifacts and a blurring, which increases towards the center of the reconstruction. To suppress these artifacts, each absorption profile before back projecting must be filtered in the Fourier domain to filter out the specific ranges of the Fourier components. The low-frequency components tend to be responsible for blurring, while the high-frequency for features and sharp details.
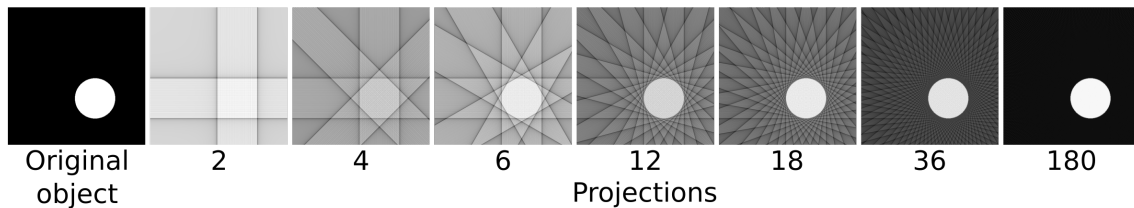
Figure 1.7.: The process behind the FBP algorithm. The transmission profile of the absorbing disk is recorded in several directions. Then, each profile is propagated back through the image space along the acquisition direction. Those regions in the image space where the back projections overlap are accumulated. The more back projections are overlapped, the closer the result of the accumulation of them resembles the original object. The reconstructed image suffering from blurring and starlike artifacts can be improved by filtering each transmission profile before back-projecting.

The efficient implementation of the Fourier transform allowed to apply the Fourier slice theorem for a rapid tomographic reconstruction. The main idea is depicted in Fig. 1.8, this theorem states that the one-dimensional Fourier transform of the projection at a given angle of a two-dimensional object is equal to the line through the center of two-dimensional Fourier transform of the reconstruction of this object. Thus, placing all Fourier transformed projections at respective angles spanning 180° onto the 2D Fourier space will fill the entire Fourier space representation of the object. Afterward, interpolating data from its polar grid to a Cartesian grid allows for recovering the object reconstruction by inverse Fourier

transforming of the interpolated data [13]. Due to fact that interpolation is a time consumable task, a range of optimizations was proposed [14].
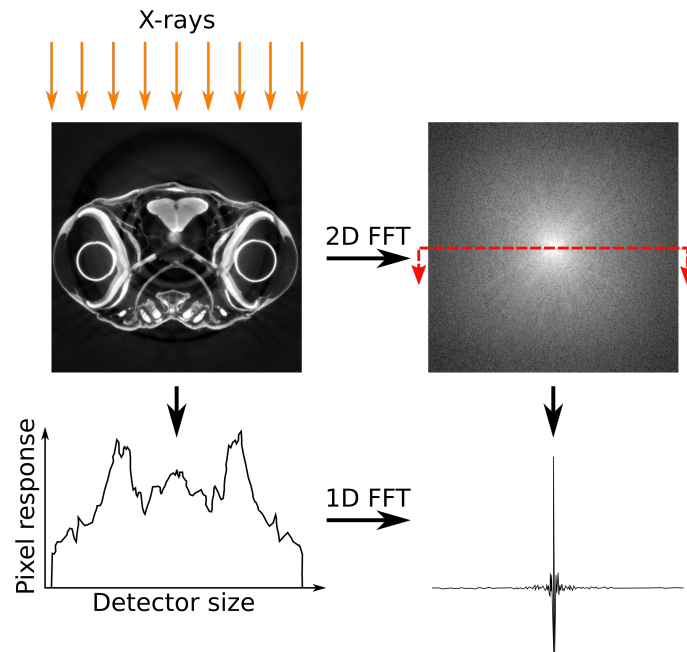
Figure 1.8.: The illustration of the Fourier slice theorem.

The stack of the reconstructed slices constitutes the 3D volume, whose resolution is determined by the dimensions of the voxels. The reconstructed volume consists of an array of these voxels on a regular three-dimensional grid, where each voxel is associated with a particular physical value, for example, the average linear absorption coefficient within that voxel.

1.3.2. Specimen preparation

The X-rays transmitting through the sample interact with its electron-poor and electron-dense regions that allows it to be visualized by the contrast of the structures against the background. In a case of biological tissues and weakly absorbing materials, they primarily consist of elements with a low atomic number, such as carbon and oxygen, providing very little contrast by imaging with hard X-rays.

Thus, in most cases, imaging of such samples enable to investigate only the outline with a complete absence of internal structures, which are crucial for the understanding of architecture and functions.

To efficiently visualize biological microstructures and architecture of weakly absorbing materials, they should be labeled with an appropriate contrast agent. During the history of usage X-rays, many contrast agents have been proposed. Among them are osmium tetroxide [15–18], gold [19–22], silver [23–26], iodine [27–30], phosphotungstic acid [28], mercury [18], lead [16], and others. Along
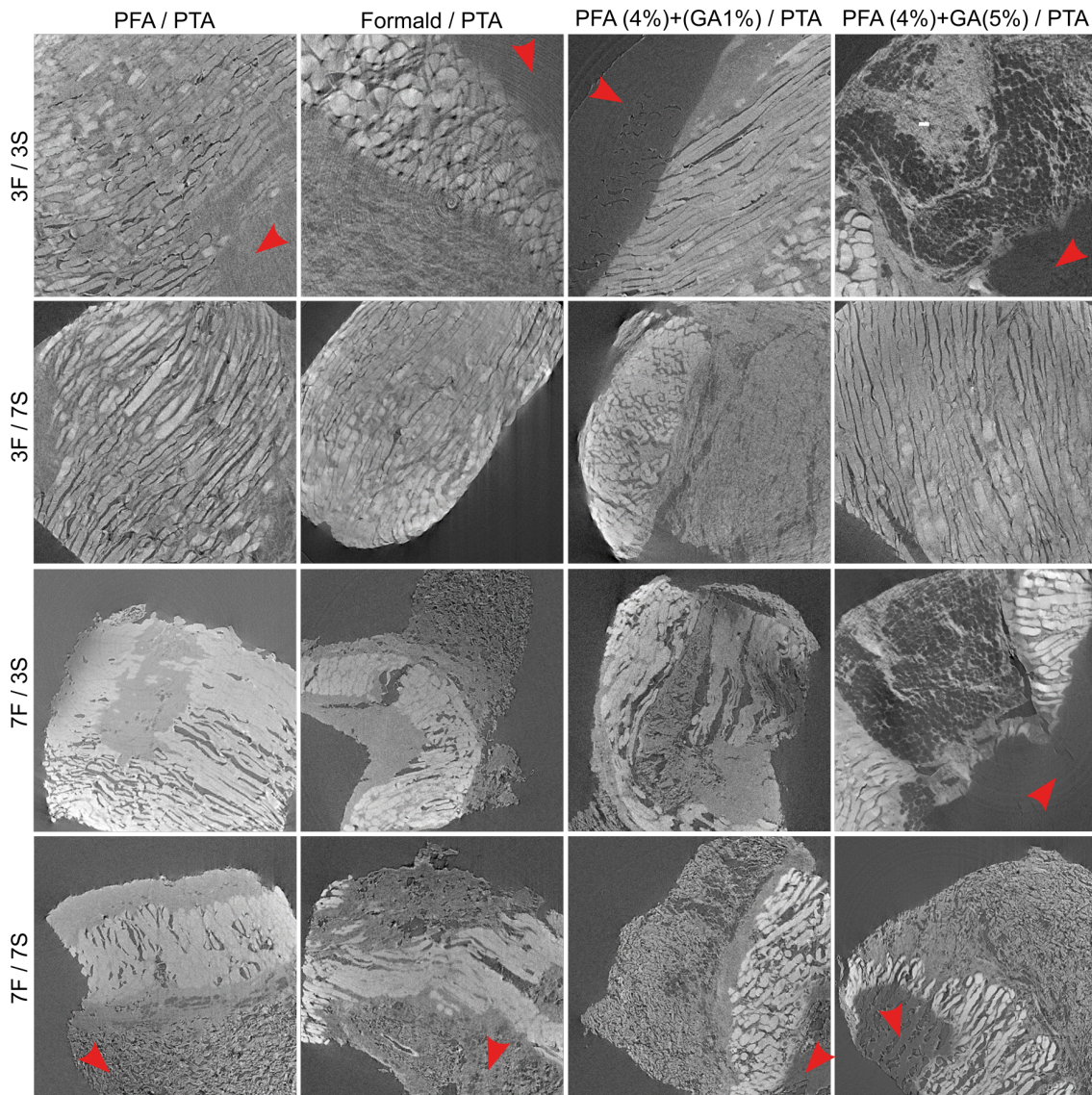
Figure 1.9.: The example of optimization sample preparation process for the hypodermis layer of mouse skin. At the horizontal axis, different variants of solutions for fixation/staining are presented, whereas the vertical axis depicts different combinations of fixation/staining duration. The red arrows point the places at samples which were poorly labeled due to an improper solution, concentration, fixation or staining duration.

with their absorptivity properties, contrast agents should be able to penetrate the whole way into a sample. Therefore, staining agents are usually chosen by taking into the account its diffusion rate, rather than specific binding properties. As an example, the systematic studies of various agents have shown that potassium iodide and mercuric chloride demonstrate fast diffusion rates [31], while phosphotungstic acid has a slower rate of diffusion, but it possesses specific binding properties efficiently applied for visualization of collagenous structures [32, 33].

Mentioned chemicals have high atomic number elements which bind differentially to internal regions of a sample. That allows significantly enhance the visi-

bility of structures due to more efficient absorption of X-rays, since it is approximately proportional to the product of electron density and the cube of an atomic number of the element, whereas the phase-contrast imaging methods rely on the phase shift [34, 35], which is proportional to only the electron density.

Every biological tissue or material has an unique staining protocol aimed at specific needs. Usage of the wrong protocol will lead to incomplete or improper staining due to differences in sample nature and composition, influencing on stain-binding affinities and diffusion rates. Therefore, a trial-and-error approach is widely applied (Fig. 1.9), which assume testing multiple stainings to determine the optimal time and solution concentration [36].

1.3.3. Imaging quality calibration

There is no general imaging setup optimized for all data quality characteristics like spatial resolution, a level of noise, acquisition time, a field of view at the same time. The imaging setup must be calibrated with a particular assurance phantom prior the experiment to ensure not only high data quality but also reproducible and comparable quantitative results with respect to other similar samples. Unfortunately, data quality parameters are not independent, that is why the setups are usually optimized only for some of the parameters while compromising others.

Spatial resolution determines the smallest distance between two distinguishable structures in an image. Nevertheless, it should not be confused with the spacing or the size of voxels of the reconstructed data. They are usually smaller than the actual resolution of the imaging setup defined by the modulation transfer function. In a case of *in vitro* imaging of biological tissues, spatial resolution can be close to the resolution of histologic sections, that is almost at a cellular level (*e.g.*, 5–10 µm). Whereas, *in vivo* imaging of small animals and objects of comparable sizes sometimes imposes the limitation on a spatial resolution to shorten the acquisition time and consequently reduce radiation dose. However, calibration of spatial resolution should be guided by resolvability of target structures.

Contrast resolution is another important parameter, which is based on contrast differences to resolve the discrepancy between structures. Contrast emerges due to differences in the energy-dependent attenuation coefficients of structures making up the sample. These coefficients are proportional to the electron density, and produced the grayscale values in the reconstructed data representing a quantitative description of the density of the material. Since contrast is inversely proportional to the energy, the higher contrast can be achieved at low photon energies. However, to find the optimal energy, many other parameters must be taken into account (*e.g.*, the quantum efficiency of the detector).

The noise level drastically influences the resolvability of structures, which in turn depends on the radiation dose, and hence on the exposure time. Although, in

some cases, contrast resolution can be traded for spatial resolution, due to high absorptivity properties of the investigated sample (*e.g.*, calcified bones).

1.4. Automated data analysis

1.4.1. Image representation

The reconstructed volume is a 3D array of voxels, which represent the linear attenuation coefficients of corresponding material at each discretized point of the volume. Every voxel can be presented as a 32-bit integer number with a value from 0 to 65535, or as a 32-bit floating number in a range from -3.4×10^{38} to 3.4×10^{38}. By such large dynamic ranges, the density of the material can be very accurately described. Modern CCD and sCMOS detectors providing a high dynamic range of 16-bit, a pixel size of several micrometers, rapid readout, and a matrix size higher than 2000×2000 pixels can easily produce hundreds of gigabytes of data per acquisition session. Thus, in the case of time-resolved or high-throughput experiments involving many samples, it causes data acquisition, handling, and segmentation challenges.

1.4.2. Image noise and artifacts

The tomographic data can suffer from different artifacts and noise from various sources. Most of these artifacts significantly complicate the further analysis, and therefore they must be suppressed as much as possible. The examples of some frequently occurred artifacts in X-ray imaging are presented in Fig. 1.10.
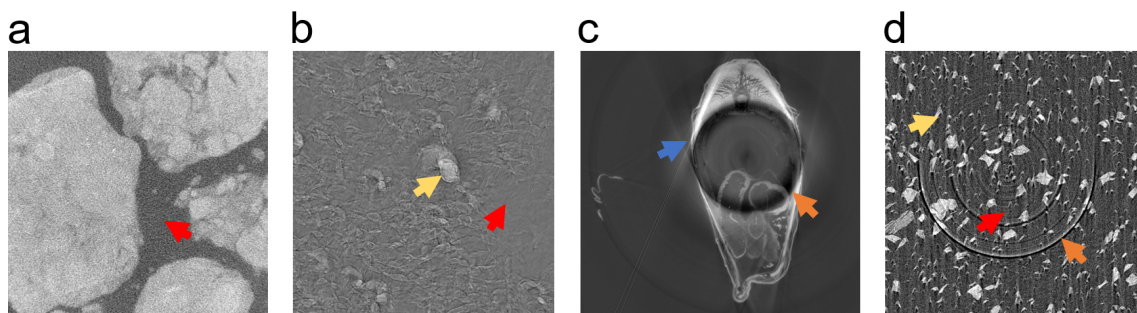
Figure 1.10.: The examples of some types of X-ray imaging artifacts in several samples: a) bioglass with large inclusions; b) composite scaffold; c) medaka fish (tail part); d) bioglass with small inclusions. The presence of a specific artifact is marked with an arrowhead of the corresponding color (red - statistical noise, yellow - motion artifacts, orange - ring artifacts, blue - streak artifacts).

Statistical noise

An insufficient photon flux caused by short exposure time or low absorptivity of a sample leads to an increase in statistical noise complicating the subsequent analysis. It decreases the visibility of sample structures and makes it difficult to distinguish adjacent parts. The noise is not constant over the volume, it usually spreads out the center of the object, due to higher X-ray attenuation of rays passing through the center. The ability to analyze the sample structures is limited by the signal-to-noise ratio (SNR) and contrast-to-noise ratio (CNR) characteristics, which quantitatively describe the detectability and distinguishability of the structures. The best achievable SNR depends on the type of sample and imaging setup configuration providing the best possible photon flux and energy.

Streak artifacts and angular undersampling

Streak artifacts can appear along straight edges of structures due to an insufficient number of acquired projections, or from the presence of small highly absorbing particles in a sample. These artifacts have a long history in various application fields of X-rays, so a vast number of approaches were proposed to mitigate them [37]. The most developed class of metal artifact reduction (MAR) methods is based on the projection completion approach, in which corrupted data inside the metal trace is refined in the sinogram domain. The correction process usually relies on data generated by interpolation techniques [38], reprojection from a prior image [39] or a combination of both that involves normalization [40]. Other MAR techniques rely on data acquisition improvement, iterative reconstruction, and image post-processing. The latter has only limited success [41] and their impact are more perceivable in conjunction with projection domain correction [42]. Recently, novel methods based on deep learning have emerged, in which a network is trained on labeled data to learn an end-to-end mapping of patches from metal-corrupted X-ray data to their corresponding artifact-free ground truth data [43].

Ring artifacts

Rings artifacts manifest as full or partial circles centered at the rotational axis of each slice of the reconstructed volume. They are caused by differences in the sensitivity or defects of adjacent detector elements [44, 45]. Since these abnormal pixel values appear at the same position on every projection, the tomographic reconstruction process generates ring patterns which reinforce towards the center due to a peculiarity of a back-projection operation. Due to the nature of these artifacts, they are much more amenable to suppression than others (*e.g.*, beam hardening). The ring artifact correction methods are usually divided into two groups: the pre-processing and the post-processing techniques [46]. The pre-processing techniques mainly work in the sinogram domain, where rings are represented

as line artifacts and can be suppressed by filtering the Fourier or real space before the tomographic reconstruction [47–52]. As well, the ring removal process can be integrated into the iterative tomographic reconstructions algorithm [53]. Recently, a new method based on the helical acquisition mode was proposed to reduce the effect of systematic errors from detector elements by transforming slice-wide ring artifacts to volume-wide helix-like artifacts reducing their severity [54]. The post-processing methods perform calculations in the image domain, where the reconstructed slices are converted to polar coordinates, and rings are filtered out as line artifacts [55, 56, 45]. The hyper-parameters for the ring correction algorithms should be finely tuned for the specific application, because some features may be easily lost during the correction process, or additional artifacts can be introduced.

Motion artifacts

Motion artifacts are usually caused by movements of the specimen or its structures during the data acquisition process [57, 58]. Also, it can appear due to detector movement relative to the sample. Since there is a significant discontinuity between the first and last projections, the reconstruction process produces breaks in the edges having asymmetric tails. The stable sample fixation at a tomographic stage can help to avoid this kind of artifacts.

1.4.3. Deep learning

Image classification is a fundamental problem in computer vision, whose goal is to categorize images into one of several predefined classes. As well, it forms the basis for other tasks such as detection, localization, and segmentation. Traditional machine learning methods suppose a dual-stage approach, where hand-crafted features are first extracted from images using feature descriptors, and then used as an input for a classifier. The major drawback of this approach is that the classification accuracy largely depends on the feature extraction procedure, which could be compromised by lack of experience in the domain.

In recent years, deep learning models that are composed of multiple layers of non-linear information processing for feature extraction, transformation, pattern analysis, and classification, have demonstrated great performance. The most remarkable case is convolutional neural networks (CNNs), which became the leading architecture for most image detection, recognition and classification tasks. It was greatly promoted during the deep learning renaissance, which happened due to the emergence of high-performance graphics processing units (GPUs), large datasets and new algorithms. CNNs have captured large attention for image classification tasks in the ImageNet Large Scale Visual Recognition Challenge, where the architecture of Krizhevsky *et al.* [59] has shown the highest accuracy among others.

CNNs are feed-forward networks, *i.e.*, they pass information only in one direction, from inputs to outputs. Similarly to artificial neural networks (ANNs), they are as well biologically inspired by the visual cortex in the brain consisting of alternating layers of simple and complex cells [60, 61]. The classical CNN architecture consists of convolutional and pooling layers, which are grouped into blocks. One or several fully-connected layers may follow these blocks, as in traditional feed-forward networks. The blocks are usually stacked on top of each other to compose a deep model. A schematic representation of a typical CNN architecture aimed at image classification is illustrated in Fig. 1.11. An image is fed directly to the network, where it is passed to several convolutional and pooling layers. Then, representations from these layers are transferred into one or more fully-connected layers. In the end, the last fully-connected layer produces the predictions of a class label.
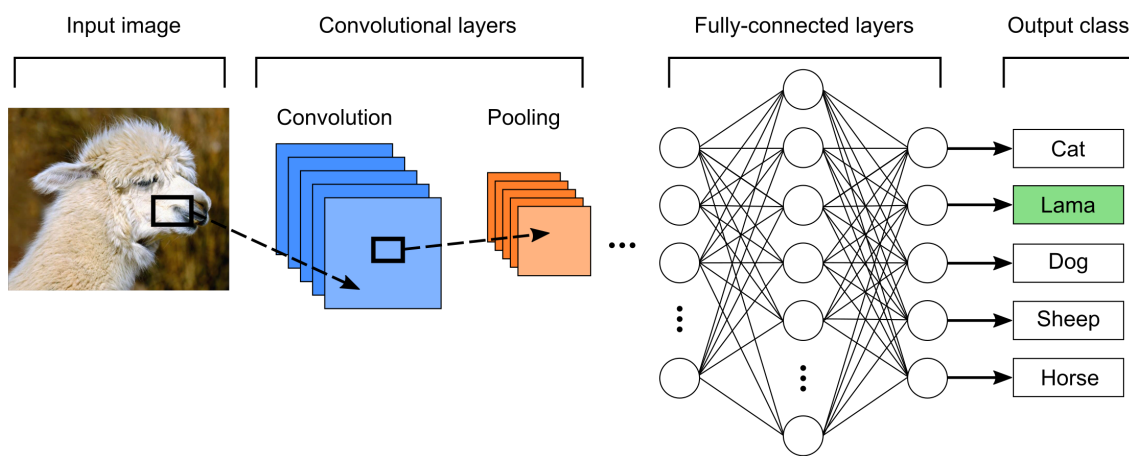
Figure 1.11.: Typical architecture of convolutional neural networks.

Convolutional layers

The convolutional layers perform feature extraction; thus they learn the feature representations of their input images. The neurons constituting the convolutional layer are organized into feature maps. Every neuron of a feature map has a receptive field, which is connected to a neighborhood of neurons at the previous layer via trainable weights, sometimes called as a filter bank [62]. Inputs are convolved with the learned weights to produce a new feature map, and the convolved results are fed to a non-linear activation function. All neurons constituting every feature map are restricted to have equal weights. However, different feature maps within the same convolutional layer have different weights; thus several features can be extracted at each spatial location [63, 62]. The convolutional layer with the k-th output feature map F_k can be defined as:

$$F_k = f(x * W_k), \tag{1.5}$$

19

where the 2D convolution operation is expressed as:

$$x[n,m] * W_k[n,m] = \sum_{i=-N_w/2}^{N_w/2} \sum_{j=-N_w/2}^{N_w/2} x[i,j] \cdot W_k[n-i, m-j], \qquad (1.6)$$

where x denotes the input image, W_k is the k-th feature map related to the specific filter, N_w is the size of the feature map, and f indicates the non-linear activation function [64]. The activation function allows extracting non-linearities from inputs. Initially, traditional sigmoid and hyperbolic tangent functions were actively used. However, in recent time rectified linear units (ReLU), expressed as:

$$f_{\text{ReLU}}(x_k) = \max(x_k, 0), \qquad (1.7)$$

have gained popularity due to its unique properties [65]. It keeps only the positive part, whereas the negative part is reduced to zero that allows introducing sparsity and be resistant to the vanishing gradient problem.

Pooling layers

The goal of pooling layer is to decrease the spatial resolution of the input feature maps and to introduce invariance to translations and possible distortions [63, 62, 66]. Initially, the first architectures of CNNs employ average pooling layers to propagate the average value of a small neighborhood of an image to the next layer [63, 67]. However, in more recent architectures, max-pooling layers propagating the maximum value within a receptive field to the next layer became popular [59, 68]. The max-pooling operation finds the largest value of each receptive field and can be expressed as:

$$F_{ij}^k = \max_{(p,q) \in \Re_{ij}} x_{pq}^k, \qquad (1.8)$$

where the output of pooling operation over the k-th feature map is denoted by F_{ij}^k, x_{pq}^k indicates the element at a spatial location (p,q) within a pooling region $\Re_{ij}$, which is a receptive field centered at the location (i,j) [64]. The intuitive description of max-pooling operation is presented in Fig. 1.12. In this case, a max-pooling operation will be performed over the entire image searching for the maximum value in non-overlapping local regions of 2×2 pixels and will produce the output image of 2×2 pixels consisting of maximum values found in the respective local regions.
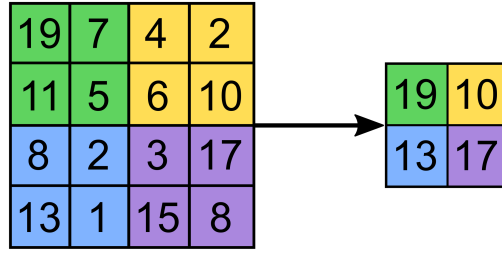
Figure 1.12.: The max-pooling operation of a filter size 2×2 pixels with a stride of 2 pixels applied to an input image of 4×4 pixels.

Fully-connected layers

The fully-connected layers usually follow after several convolutional and pooling layers, stacked on top of each other to extract more abstract feature representations of inputs. In these layers, the extracted features are interpreted and employed in training of the network by minimizing a specific loss function. It is common to use the softmax loss function for the classification problems due to its simplicity and probabilistic interpretation [59, 68]. The softmax loss function for the i-th input feature x_i with the corresponding label y_i, can be written as:

$$f_{\text{sm}} = \frac{1}{N} \sum_i L_i = \frac{1}{N} \sum_i -\log \frac{e^{f_{yi}}}{\sum_j e^{f_j}}, \tag{1.9}$$

where the j-th element ($j \in [1, K]$ and K is the number of classes) of the vector of class scores f denoted by f_j, and N is the amount of training data. In this case, f is the activations of a fully-connected layer W, thus f_{yi} can be denoted as $f_{yi} = W_{yi}^T x_i$, where W_{yi} is the yi-th column of W.

1.4.4. Workflow of data analysis for X-ray imaging

The data analysis workflows were developed to standardize and automatize the processing of datasets of a similar kind (*e.g.*, datasets of animals belonging to the same species). The workflow in the simplest case is a sequence of independent processing operations, where the result of one operation is fed to the input of another. The independence of operations allows being implemented and assembled using different software packages, libraries, and platforms, which makes such workflows are troublesome for deploying and reusing.

There are many examples of workflows developed for different application cases in various research fields, such as a study of living processes in model organisms [69–71] to understand disease mechanisms, and further apply to a human. Using modern non-invasive techniques allows for studying gene knockout animals (*e.g.*, mice [72, 73]) to gain an insight into gene function [74–76] and various insects

(*e.g.*, fruit fly (*Drosophila melanogaster*)) for understanding of embryogenesis [77, 78], behavior patterns [79] and brain morphology [80, 81].

All considered application cases implement custom automated analysis workflows designed to answer particular research questions and tailored for specific data modalities starting from μCT [70, 74, 76] and light microscopy [69, 71, 77, 80, 81] to conventional photography [79]. Nevertheless, most of them have a similar structure composed of several of the following stages: pre-processing, sample alignment, localization, segmentation, and morphological analysis. Moreover, due to the generality of image processing principles in these stages, they are equally applicable to data from different modalities but may require some adjustments (*e.g.*, the alignment process may use specific fluorescence expression patterns as in [80]).

Data pre-processing

The pre-processing is an important part of the automated analysis workflow. The signal acquired from a detector usually suffers from noise and artifacts of various kinds, which should be weakened or removed before transmitting data to the next stage.

The nature of artifacts depends on the acquisition setup because different ways of imaging suffer from different kinds of artifacts and noise sources. The fluorescent microscopy techniques can produce sequences of multi-channel images forming a 3D stack, which usually suffer from such artifacts as non-uniform illumination, blur, statistical noise and can be corrected with a number of approaches [82–87].

The μCT laboratory sources allow acquiring a series of projections, which are then reconstructed into a 3D volume. The reconstructed volumetric data suffer from a range of artifacts occurred due to inappropriate parameters of an imaging setup, an acquisition device, sample composition, and size. If a sample is composed of several different materials, beam hardening artifact may occur due to selective attenuation of lower energy photons of a polychromatic beam. Hot pixels at a detector or cracks at a scintillator lead to ring artifacts, which can be partially removed with methods mentioned in Sec. 1.4.2. Streak artifacts occur due to the presence of highly absorbing parts in a sample surrounded by weakly absorbing material; however, such artifacts may usually be found in the medical imaging, where there are prostheses or implants. The X-ray beam instability can be a cause of contrast differences in neighborhood projections; thus it should be corrected with flat-field correction methods [88, 89]. An insufficient angular sampling of projections could also result in streak artifacts at straight parts of a sample. To cope with this, the slices can be obtained with algebraic reconstruction algorithms, which perform iterative reconstruction by modeling the forward and backward projection process and allow to take into account various prior knowledge to improve the reconstruction quality [90, 91].

The data produced by SR-µCT suffer only from a part of laboratory µCT artifacts (Sec. 1.4.2) because the beam has much better properties in comparison to the lab sources (Sec. 1.2.3). Besides the specific artifacts and noise introduced by the acquisition systems, there is a common way to improve contrast and clarity of data. As an example, it is a histogram equalization [92], non-linear filtering which allows preserving edges of structures [93, 94], and local contrast enhancement [95–97]. To reduce the size of data and complexity of structures, one can utilize a coarse-to-fine approach by multi-level downscaling with subsequent Gaussian smoothing [98].

In some cases, the data can be acquired in 2D or 3D chucks. For instance, the result of studies using microscopy is a bunch of 2D images, because the sample could be too large for the field of view, so series of overlapped images are created. In experiments studying elongated samples with SR-µCT, whose length is much larger that height of the field of view, the samples are scanned in chunks by shifting along the Z-axis per the data acquisition session. Then, these 2D or 3D datasets should be stitched into a single image or volume to be passed further. The stitching algorithms were initially used for creating panoramic views from sets of overlapping photos [99], and later were adapted for scientific imaging [100–102].

Sample alignment

The primary goal of the sample alignment is to align all datasets involved in the analysis process to a common coordinate system defined by the reference dataset or the manually arranged set of landmarks. Hence, this stage is similar to the rigid intra-modal registration, which operates with datasets from the specific modality and composed of rotation and translation geometric transformations describing sample displacement. The proper alignment is vital for the successful analysis because the segmentation algorithms are tailored to work better with the datasets having the particular orientation; otherwise, the behavior could be unpredictable. Moreover, the analysis at the substructure level requires accurately extracted sample structures by correctly determined bounding boxes, which is impossible when datasets are not aligned. The demand in alignment usually occurs in cases when the sample cannot be reliably fixed during the data acquisition process, which results in that all scanned samples are presented in the datasets in different spatial positions that complicates the further analysis.

In the medical imaging, this procedure is rarely needed since the doctor usually guides a pose of a patient. In the pre-clinical research on the rodents and other vertebrates with the laboratory µCT systems, they are typically mounted on the special stage, which guarantees that all specimens are positioned in the same way, and no additional alignment is required. In the scientific imaging of living organisms and materials, samples sometimes cannot be properly oriented relative to each other. In the experiments aimed at studying of insects with the SR-µCT, insects are placed into plastic containers filled with viscous solution helping to

avoid dehydration and fixing them in the same orientation. However, it is impossible to set all samples in the same way due to morphological differences, so additional pose adjustments are required. Therefore, the sample alignment is the essential part of the analysis workflow, especially in cases when samples cannot be reliably positioned.

The registration algorithms are divided into two groups, namely the landmark-based methods which utilize points and surfaces extracted from datasets, and the intensity-based methods using voxel intensities in the iterative registration process.

Landmark-based techniques   These methods derive the geometric transformation using features extracted from the data, represented as landmarks, surfaces or point clouds.

The landmarks are points in a target dataset identified automatically, they could be represented by an intensity maximum, gradient, or other computable features. The most frequently used feature detectors are SIFT [103–106], SURF [107, 108], and MSER [109]. The point-based registration methods use these feature detectors to identify similar points in the target and reference datasets to derive the spatial transformation by minimizing the Euclidean distance between these points [110, 111].

The surfaces in some cases can be even more distinct than landmarks. The iterative closest point algorithm consists of two stages, namely, identifying for each point of the target surface the closest point on the reference surface, and then finding the spatial transformation of these point sets [112–114]. The algorithm iteratively redefines the closest point set and continues until it finds the match between these surfaces by minimizing the cost function based on the Euclidean distance.

Intensity-based techniques   These methods rely on voxel intensities of the target and reference datasets to derive the geometric transformation required to perform the alignment. The most popular similarity measures are based on difference and cross-correlation of voxel intensities and information theory.

The measures based on the intensity difference usually employ the sum of squared differences (SSD) [115, 116]. The cross-correlation (CC) measure is based on the assumption that there is a relation between the intensities of the corresponding structures in the datasets [117–119]. The SSD, the CC, and derivatives of these similarity measures are applicable for intra-modal registration.

The similarity measures employing the information theory are usually based on the mutual information (MI) [120–122]. The MI is based on the Shannon entropy which is estimated from the joint probability distribution of the dataset voxel intensity. This measure estimates how well one dataset explains the other, *i.e.*, it is based on the assumption that there is a function between the intensities of both

datasets. The MI is applicable for both intra- and inter-modal registrations and maximized when the datasets are properly aligned [123–125]. There were also introduced other measures, *e.g.*, Rény's entropy [126], Tsallis entropy [127–129] and Havrda-Charvat's entropy [130].

Localization of structures

Automated localization sometimes may be critical for initialization of automatic segmentation methods and further analysis [131–136], image classification [137], and content-based image retrieval [138, 139]. The localization is a process of determination of the region in a data containing the target structure, whereas the segmentation is aimed at accurate delineation of this structure.

The localization of structures has an interest for many cases, where the specific parts of a sample should be studied separately, *e.g.*, organs of living organisms, inclusions and defects in materials, components of integrated circuits, cells grown on a surface of biomaterials, and others. In some cases, it plays an essential role in such tasks as segmentation and registration. Moreover, it enables faster data inspection and visualization of target structures. In the field of medical imaging it helps in treatment planning, for example in radiotherapy, a volume of an organ is required to compute a radiation dose [140]. In the material science, the analysis of material damage involves localization to analyze shape evolution of cracks and inclusions. The result of such analysis is further used in finite element modeling and applied in various industries. Also, it is important for high-throughput experiments which require automated analysis workflows for detailed analysis of specific parts of a sample especially is the case of complex heterogeneous samples.

During the history of multi-dimensional data analysis, only a few dedicated methods for automated localization of structures in composite materials, organs or other structures were proposed. These methods can be divided into three groups: knowledge-based, atlas-based and learning-based approaches.

Knowledge-based approaches The spatial relationships between sample structures can be represented by a structural model, which is a graph where the nodes are the structures, and the edges are their relationships. This representation in conjunction with organ-specific appearance models was used to localize organs in a human body [141]. Fouquier *et al.* [142] proposed to localize human brain structures in pre-computed sequential order, following *a priori* knowledge about the spatial relationships between these structures. However, due to the sequential nature, this process tends to propagate errors and therefore requires backtracking. The solution for this issue was proposed in [143] by performing global localization which takes into account a constrained network derived from

the structural model. Even though this approach relies on robust *a priori* knowledge, it is computationally expensive and requires thorough parameter tuning, which makes it difficult to apply in practice.

Atlas-based approaches The atlases are datasets with known positions of target structures. The atlas-based approaches are widely applied in segmentation, but also can be used for localization purposes. They are based on a registration process, whose goal is to find an optimal transformation to warp atlases to a dataset with unknown positions of target structures. By placing all atlases into a common space allows to roughly localize the target structures [144–148]. These approaches were proven to be robust techniques, especially if many atlases with various textural and shape variations of target structures are available. The diversity of presented variations allows to perform more accurate registration and thus localization by selecting only the most similar atlases to the studied sample [149–151]. The atlas-based techniques usually require exhaustive parameter tuning and are considered to be slow due to the computational complexity of the optimization process, which is a core of any method employing registration.

Learning-based approaches These approaches are based on machine learning algorithms trained on a subset of data to predict coordinates of the bounding box containing the target structure. In the work of Zheng *et al.* a dedicated localization method using marginal space learning (MSL) was introduced and used to localize organs in medical CT datasets [152]. The bounding box was determined in three successive stages: finding its location, orientation, and size. This strategy allows to significantly reduce the parameter space at each following stage while allowing to calibrate the parameters found at the previous stages. At every stage a set of candidate bounding boxes are generated, then the optimal one is chosen using a probabilistic boosting tree. This method was validated and evaluated with localization of ventricles of the human heart in 3D CT and 2D MRI datasets [153].

The method allowing to localize multiple target structures simultaneously was firstly proposed by Criminisi *et al.* in [154]. The localization was formulated as a multivariate regression problem and solved with a regression random forest. The regression forest determines the spatial positions of the six bounding walls around the target structures by textural features calculated in their neighborhood [155]. The method was evaluated for the localization of 26 organs in full-body human medical CT datasets. Thereafter, this approach was improved in the work of Gauriau *et al.* by cascading the initial regression forest with organ-specific forests [132]. This improvement reduced the localization error, but at a cost of increased execution time.

The method introduced by Zhou *et al.* consisted in defining 2D bounding boxes at image slices from three orthogonal planes [156]. Then, they were determined

with AdaBoost and an ensemble of stump classifiers trained at calculated Haar-like features. Afterward, these predicted 2D bounding boxes were combined into 3D bounding boxes by a major voting algorithm. The evaluation of this method was done for the localization of five human organs in medical CT datasets. In comparison to other mentioned methods, this one is slower and localizes only one target structure at a time.

CNNs emerged at the end of 80th [63]; however, they did not become widespread due to high computational complexity but were recalled recently [62]. These networks learn features directly from the raw data, reducing the semantic gap introduced by hand-crafted features and drastically shorten the time required to engineer features. At the moment, many works aimed at organ detection are based on the different kind of neural networks. The autoencoders were used to find rough locations of organs by detecting landmarks [157]. Various CNN architectures were designed to accurately localize organs in medical CT datasets of a human body, employing information from different axial planes [158, 137, 159–161]. The CNNs trained at slices from different axial planes perform multi-structure localization by deriving 3D bounding boxes based on slices indices where organs were detected.

Data segmentation

The goal of segmentation is the delineation of target structures, or in other words, identification of a set of voxels making up the contour or the interior of the target structure. The segmented structures of studied samples allow for quantitative analysis of various parameters related to volume and shape, which enable to perform phenotyping analysis over studied living organisms or to find optimal fabrication parameters in large-scale material optimization studies. Besides automated analysis workflows, an accurate segmentation greatly simplifies the process of manual inspection of the sample for the operator.

Currently, with the help of specialized software providing segmentation capabilities, the experts in the domain can accurately perform delineation of the target structures. However, the segmentation process of volumetric datasets composed of thousands of slices in a slice-by-slice manner is laborious, incredibly tedious, time consumable and expensive. Moreover, the manual annotating is subjective and suffer from low reproducibility that would lead to high inter-observer variability, as the segmentation quality highly depends on the individual experience and knowledge. To the end, it is physically impossible to provide manual segmentation for results from high-throughput experiments, generating hundreds of multi-dimensional high-resolution datasets. Thus, automated segmentation methods are highly demanded, especially in cases when the amount of produced data is impossible to process manually in a reasonable amount of time.

Automated volumetric segmentation is an exceptionally challenging task in the case of complex heterogeneous data. The appearance and shape variations of

target structures can significantly vary among the studied samples. The boundaries between target structures or surrounded environment are usually ambiguous with limited contrast, which can be caused by similarity in physical properties of materials making up these structures.

Basic approaches   The global thresholding is the most widely applied segmentation algorithm, whose idea is to determine an optimal thresholding value to separate regions of interest by analyzing the histogram. Depending on the approach used for determination of a thresholding value, algorithms may be divided into several groups, namely, based on analysis of the histogram shape, correlation of entropies estimated from the background and foreground pixels, correlation of the grayscale and segmented image properties or higher-order probability distributions and spatial relationships between pixels.

The region growing assumes that the pixels constituting the object of interest are connected and similar enough. A specific algorithm iteratively searches for all pixels neighboring to a current region of interest (ROI) and uses a particular selection criterion based on similarity of the intensities of neighboring pixels to the mean intensity of the region to decide on addition these pixels to the ROI.

The watershed algorithm perceives an image as a topographic surface, where low and high intensities denote hills and valleys, respectively. The algorithm starts to fill every isolated valley with water of different color (label). As the level of water rises, at some time point, water from different valleys starts to merge. To prevent this, barriers are built in the locations where water merges. The algorithm continues to fill the valleys and building barriers until all hills are under water. Finally, the segmentation result is represented as different water pools (labeled groups of pixels) separated by the barriers (background pixels) from each other.

The probabilistic fuzzy clustering groups the pixels based on their intensity and spatial connectivity with others. The algorithm iteratively tries to find a stable configuration of clusters. Once it is converged, the segmentation task is simplified to thresholding of the fuzzy connectedness values.

Model-based approaches   These approaches gained huge popularity for multi-structural segmentation in various imaging fields. A shape model or a probability atlas is estimated by averaging shape or location priors of multiple spatially aligned atlases. Constructed statistical shape models or probabilistic atlases can provide *a priori* knowledge about the location and shape of the target structures for the segmentation methods.

The statistical shape models were introduced in the early 90s by Cootes *et al.* [162]. Later on, they were improved and successfully applied in the segmentation of human abdominal organs [163], such as a liver [164], a pancreas [165], and kidneys [166] in medical CT datasets. To improve robustness to shape variability of organs, a generalized multi-resolution hierarchical shape model was proposed

[167]. Since statistical shape models or probabilistic atlases are built from a specific population of atlases matching the appearance of the target structures, these models lead to a low specificity with respect to individual appearance. The generality of such average models may impede the segmentation due to substantial inter-subject variability. To cope with these shortcomings, approaches based on subject-specific shape models [168] or subject-specific probabilistic atlases [169, 165, 170] were proposed. In recent works, spatial a priori knowledge of different abdominal organs was incorporated to facilitate multi-organ segmentation [171–176, 170, 177].

In earlier studies, Aljabar *et al.* showed that the segmentation accuracy of multi-atlas based methods highly depends on atlases selected for the target dataset [178]. Most atlas selection methods measure the global similarity between the target and atlas dataset. More advanced methods transfer global similarities into a manifold and perform the atlas selection from the learned manifold [179, 180]. However, global similarities might not adequately represent similarity due to large organs may occupy most of the space. To overcome this drawback, a region-wise local atlas selection strategy was proposed by selecting atlases for each local region [181, 182, 170]. As an example, Tong *et al.* proposed a patch-based segmentation framework to perform abdominal multi-organ segmentation in MRI datasets [183].

Machine learning-based approaches   Recently, machine learning algorithms such as random forests (RFs) were extensively used to tackle the problem of multi-structural segmentation. The RF algorithm was firstly proposed by Breiman *et al.*, and designed to handle multi-class problems and be computationally efficient in processing of large amount of 2D and 3D features based on voxel intensity and local relationship information [184]. It was widely applied in the biomedical and computer vision domains and lately adapted to the medical image segmentation. The RFs gave impetus to the development of other forest-based algorithms. The vantage point forests were originally proposed for high-dimensional data clustering and accelerated search in metric space [185]. Further, they were successfully applied for supervised image classification with 3D BRIEF [186] providing better robustness than the long-range context features [187] or SIFT vectors [188]. The random ferns method proposed by Yaqub *et al.* use so-called ferns features composed of a series of binary features representing a patch in a vicinity of a given voxel [189]. Jin *et al.* suggested combining RFs and random ferns to take advantage of both methods [190]. Richmond *et al.* presented cascaded RFs interleaved with inference in Markov random field (MRF) models for both segmentation and localization [191]. Bieth *et al.* proposed to use vantage point forests using local context features of the target structures with further gradual refinement by regional context information [192]. Zografos *et al.* proposed a method based on a boosted tree classifier trained on contextual features extracted from the multi-scale supervoxels [193]. Glocker *et al.* suggested joint classification-regression forests trained on class and spatial consistency ex-

tracted from structural information of labels [194]. Montillo *et al.* developed a novel entangled decision forests exploiting spatial correlations between the target structures [195]. The mentioned approaches were successfully applied for segmentation of various human abdominal organs in medical CT datasets.

Despite the efficiency of forest-based methods, the hand-crafted features are limited in representation capabilities of huge appearance variations of the target structures. Moreover, they cannot take full advantage of the 3D spatial information presented in the volumetric data to achieve high segmentation accuracy.

Deep learning-based methods Deep learning-based methods were emerging as a competitive branch which provides alternative approaches to traditional image segmentation tasks. The CNNs achieved great progress in 2D scientific image segmentation [196–198]. However, it is still challenging to segment target structures from volumetric CT datasets possessing different spatial properties.

In the architectures of the first CNNs such as AlexNet [59] and VGGNet [68], the last layers are usually fully-connected. This fact restricts the network to take input images of fixed size and produce non-spatial predictions, usually expressed as a probability vector of a length equal to the number of target classes. Soon after the development of first CNNs, Ciresan *et al.* used a patch-based approach allowed to make predictions with spatial information [199]. However, such strategies are computationally expensive and therefore are not directly applicable to volumetric segmentation, which requires dense prediction for every voxel in a dataset.

These fully-connected layers can be converted into convolutional layers by reorganizing the parameter weight matrix into high-dimensional convolution kernels [200]. Thus, after that transformation, the entire network forms a fully convolutional architecture, where all layers are convolutional or pooling, and both have no restriction on the size of input data. In other words, the network is capable of taking volumetric data of arbitrary size, and produce the classification probability volume of the same size. Therefore, the fully convolutional network drastically reduce redundant computations occurred due to overlapping inherent to the patch-based methods. CNNs demonstrated high performance in many challenging medical image analysis tasks, in particular for classification [201–203] and segmentation [199, 204] in the last years. The CNN-based segmentation methods can be divided into two groups, namely 2D and 3D CNNs.

The methods based on 2D CNNs perform segmentation in a slice-by-slice fashion for processing volumetric CT or MRI datasets [205, 206]. Ronneberger *et al.* proposed the novel architecture called U-Net providing a dense image-to-image segmentation [207].

Later on, 3D CNN-based methods emerged since, despite high segmentation quality of 2D methods, they still do not take into account potentially useful 3D spatial information of volumetric data. In the beginning, Roth *et al.* proposed

a pseudo-3D approach employing 2D patches extracted from orthogonal planes [208]. Afterward, pure 3D CNN-based segmentation methods were developed [209, 210] and allowed to use full advantages of volumetric data, demonstrating high segmentation accuracy of lesions in multi-channel MRI datasets. Meanwhile, various extensions of the U-net architecture for volumetric data were proposed, such as 3D U-Net [211], V-Net [212] and VoxResNet [213].

The segmentation results provided by CNN-based methods usually may be refined with classical segmentation methods, deformable models [214] or graphical models [215, 216].

Morphometric analysis

The extraction of metrics allows concluding on the current state of the sample composition. In the medical imaging it allows making decisions on the diagnosis or changing the treatment strategy. The morphometric properties allow establishing correlations of the specific genes with the anatomy changes of the organism, which helps to decipher the genome and describe the phenotype.

Object shape and distribution estimation   The shape properties of objects in a 3D space can be described with volume, surface, inertia moments, and kurtosis measures estimated from the segmentation result. The distributions of particles or small parts of the target structure can be performed with a group of methods aimed at the labeling of non-adjacent groups of voxels [217].

Orientation analysis of fiber structures   The task of orientation analysis arises in various application fields, and the way of analysis mainly depends on the nature of target structures. The orientation of amorphous structures is usually estimated over the area, whereas the fibrous structures are along the medial fiber axis. Nevertheless, these cases share similar algorithms due to the natural similarity of the estimation process. However, the quantitative information about the orientation of fibrous structures has a particular interest, since it allows to improve mechanical properties of materials and guide a fabrication process to produce a material with desired properties.

Various approaches aimed at 2D orientation analysis emerged in the last decades. The Hough transform was utilized for analysis of collagen fibers [218], electrospun polyacrylonitrile fibers [219], and alignment quantification of structures in textile composites [220]. Another approach is based on computing an intensity gradient and allows to determine the orientation at the particular point via a ratio of gradient magnitudes. It was used to quantify the orientation of cytoskeletal fibers [221] and myofiber disarray [222], fibers in human ligament fibroblast [223], unidirectional fiber-reinforced polymers [224], and collagen fibers [225]. The next

technique involves the analysis of the 2D Fourier spectrum, which enables to reveal a global orientation of structures. It was applied to quantify the orientation of nanofibrous layers of textile materials [226, 227], fibers in electrospun materials [228–230], and collagen fibers [231, 232].

Quantification of 3D orientation is presented by a few approaches, which are mainly operated in a vicinity of every voxel. The most popular approach is based on eigenanalysis on the second-order structure tensor calculated for every voxel neighborhood [233–235] for characterization of nonwoven fabrics materials, fiber-reinforced composites, and polymer scaffolds with fiber structure. A similar method estimates 3D fiber orientation from a series of 2D measurements [236–238] for analysis of reinforced composite materials. Other methods utilize 3D inertia moments [239] and shape of fibers [240] to estimate orientation in biological tissues.

Even though a lot of methods for fiber orientation analysis were proposed, there were no notable works devoted to the orientation accuracy evaluation of methods on ground-truth datasets.

1.4.5. Available software solutions

The analysis workflow of sophisticated heterogeneous data produced by high-throughput experiments usually consists of several stages such as I/O operations, pre-processing, analysis, characteristics quantification, and visualization. The composition of the workflow, as a rule, is dictated by research goals, data acquisition setup, nature of acquired data, and other peculiar properties inherent to the specific experimental setup. Nowadays, high-throughput experiments produce terabytes of data per acquisition session, which cannot be processed manually or semi-automatically [241]. Therefore, the software packages providing opportunities for building custom workflows and deploying them on distributed parallel computing environments are required to handle such amount of data efficiently.

In the past decade, a vast number of commercial and open-source software solutions were proposed. The commercial solutions such as MetaMorph, Amira (FEI), Volocity (PerkinElmer), SlideBook (3i), and ZEN (Zeiss) are usually offered as complementary tools by companies selling imaging instrumentation. Among the open-source solutions, which allow general-purpose analysis of 2D/3D data and creating custom workflows are Icy [242], ImageJ [243], CellProfiler [244], 3D Slicer [245], ilastik [246], MITK [247] and workflow systems like KNIME [248], WEKA [249], XPIWIT [250] which allow assembling workflow from independent processing blocks written from the scratch or reused from other software. These software packages were widely applied in various research fields, for example in the context of fruit fly [251] and zebrafish [252] research.

However, despite availability and popularity of these packages, most of them have limited support for deep learning technology which is rapidly gaining weight in all scientific and medical imaging fields and allow efficiently utilize a large amount of data in segmentation. Only a few packages, namely NifTK [253] and Slicer 3D (via the DeepInfer plugin [254]) provide some functionality for integration of deep learning into the workflows.

At the moment, general-purpose deep learning software is under extensive development. Due to high computational demands for training deep learning models and the complexity of efficient usage of the modern hardware, a vast number of deep learning libraries and platforms were developed and adapted for the variety of programming languages, such as cuDNN [255], TensorFlow [256], Theano [257], Caffe [258], Torch [259], PyTorch [260], CNTK [261], and MatConvNet [262]. These platforms and libraries facilitate creating sophisticated deep networks providing high-level of programmatic abstractions hiding all nuances of parameter optimization procedures required for a training process, and providing efficient implementations of routines involved in the process of network creation. These frameworks were developed to be extremely flexible, optimized and providing a general interface for deep learning technology, therefore it is challenging to use them directly. This issue inspired other developers to create platforms simplifying the process of network creation for common usage scenarios, such as Keras [263], and TensorLayer [264] for TensorFlow and Lasagne [265] for Theano. Although, the proposed platforms remained to be general-purpose without focus at any specific research field. However, recently NiftyNet [266] for TensorFlow was proposed, this platform facilitates application of deep networks in medical image analysis providing all necessary functionality from data loading to model evaluation. In the same time, the Deep Learning Toolkit was developed by Pawlowski *et al.* [267] providing building blocks for fast prototyping of deep learning models and modules in the context of medical image analysis.

The segmented data produced by a software package or library should be quantitatively estimated using the morphological analysis approaches. However, among the considered aspects of morphology analysis, only analysis of fiber-like structures presents a substantial variety of techniques and software. There are various proprietary and open-source solutions available to conduct fiber analysis in 2D or 3D. Among the proprietary software, FEI Avizo 3D with Fiber Analysis module, the Automated Fiber Analysis included in the software shipped with Phenom Scanning Electron Microscopes, and VGStudio MAX with its Fiber Composite Material Analysis Module, are commonly used. Also, a vast variety of open-source solutions is available, most noticeable are the stand-alone application FiberScout [268] which is mainly aimed at the 3D analysis of fiber-reinforced materials for industrial CT, and plug-ins for ImageJ [243]. These plug-ins include DiameterJ [269], FibrilTool [270], and OrientationJ [271] which offer the analysis of 2D images acquired with various imaging techniques like light and electron microscopy, μCT, and others. The considered software solutions are mostly oriented only to a single aspect of fiber characterization; thus a complete analysis can be performed only with multiple software tools. Moreover, a lack of inte-

gration capabilities, inability to process both 3D and 2D data equally well and limited parallelization capabilities impede efficient implementation of automated analysis workflows.

1.5. Hypotheses and open questions

Despite that previous research on approaches to X-ray data analysis provide a broad range of solutions, they initially were designed for ad-hoc scenarios in the medical imaging domain employing devices which are not optimized for throughput and produce relatively low-resolution CT data. Thus, these approaches cannot be directly applied to tackle the problem of high-resolution μCT data analysis of model organisms and materials due to vast discrepancy in data characteristics and analysis process specificities; hence, there are several unsolved questions:

- Existing automated analysis workflows are rigid and mostly aimed at studying of particular specimens in the experiments tailored to the specific data acquisition setups and producing a relatively small amount of data; therefore, they are not applicable for high-throughput X-ray experiments for various specimens and producing terabytes of μCT data per acquisition session.

- Most available data segmentation approaches are mainly developed and applied in the medical imaging field where low-resolution data acquisition systems are used. Thus, they provide very limited applicability to in-depth hierarchical analysis of high-resolution μCT data acquired with modern synchrotron light and X-ray laboratory sources.

- Existing methods aimed at fiber orientation analysis are usually based on the second-order structure tensor, which has high computational complexity limiting the applicability of such methods to high-resolution large μCT datasets by requiring downscaling or analyzing only the specified ROIs.

- Most of the automated analysis workflows currently used in X-ray experiments are composed of independent software packages. Usage and maintenance of such workflows are complicated due to imposed dependencies on the third-party packages and libraries.

- Even though many fiber analysis approaches have been presented in the last decade, their programmatic implementations are not readily available to the end users, suffer from insufficient parallelization, require the excessive amount of memory, and are not adapted for large μCT datasets that impedes integration with the automated analysis workflows.

1.6. Objectives and thesis outline

According to unsolved problems stated in the previous section, the primary objectives of this thesis are:

1. Design the automated analysis workflow capable of handling a large amount of heterogeneous µCT data produced by acquisition setups based on synchrotron light or laboratory X-ray sources involved in the high-throughput experiments aimed at studying model organisms.

2. Develop an efficient and reliable segmentation approach for the hierarchical analysis of high-resolution µCT data. The method should be able to perform rapid and accurate segmentation of substructures presented in µCT datasets of hundreds of gigabytes.

3. A rapid and accurate 3D orientation analysis method applicable to large µCT datasets and executable on modern parallel computing environments.

4. The performance analysis of the designed workflow over the generated phantom datasets to determine the optimal parameterization and limitations of each stage.

5. Develop the algorithm for generating a synthetic dataset of the specified size, noise severity, number and parameters of included fibers.

6. The accuracy and throughput analysis of the proposed orientation estimation method in comparison to another approach on the synthetic dataset in various scenarios of fiber composition, data size, and noise severity.

7. The implementation of all stages of the proposed workflow as independent modules, using the state-of-the-art programming libraries, to make possible integration into any workflow management library or package.

8. The implementation of the fiber analysis module, which provides capabilities for orientation estimation and generation of synthetic datasets.

9. Apply the developed workflow to several real-world application cases from the life and material sciences.

To make possible accurate and efficient analysis of large µCT data produced during high-throughput X-ray experiments, the concept of the automated analysis workflow along with all thoroughly evaluated and validated intermediate stages is presented in Chap. 2. The novel method for fiber orientation analysis capable of processing large µCT datasets by taking advantage of modern parallel computing environments is described in Chap. 3 along with the comparison to another popular method. Besides the systematic validation of the developed workflow on the phantom datasets, it was also applied to facilitate answering the real-world scientific questions as illustrated in Chap. 4. Such questions as the existence of dependencies between the fish strain and distinct morphometric properties of the organs of medaka fish (*Oryzias latipes*) and analysis of polycaprolactone 3D

scaffolds, which are widely applied in the field of tissue engineering, to find the optimal fabrication parameters. Finally, a summary of the completed tasks and possible improvements are discussed in Chap. 5.

2. An automated machine learning-based segmentation framework for X-ray data

2.1. Overview and related work

The efficient data analysis is a crucial part of high-throughput experiments involved in studying a large number of samples of a similar nature. The data analysis workflows are widely applied in various fields, *e.g.*, in the life sciences, they are used for studying of living processes in model organisms [69, 70] to understand physiology and diseases of complex living systems with perspective to apply obtained knowledge for treating human diseases. As it was already mentioned in Sec. 1.4.4, the workflows usually consist of several interconnected processing stages.

Most workflows operate with single images of relatively small size, which are easy to transfer over a network to a remote server for further processing. However, transferring of μCT datasets may be problematic due to large size up to dozens of gigabytes. The fault-tolerance in processing single images is not crucial, because usually, images are independent, and analysis failure of any image will insignificantly influence results. Whereas, in the case of μCT datasets composed of sets of slices should be treated as single 3D images because analysis failure of any slice will violate sample integrity. The workflows considered in Sec. 1.4.4 do not provide capabilities for hierarchical analysis, which is in high demand for handling large datasets describing complex multi-structural samples. They assume that the complete dataset should be processed at once, without data reduction and narrowing the location for processing. Also, these workflows do not take into account the results from previous executions, which potentially could contain useful information to improve their overall performance.

Investigation of the proposed workflows for applicability to the current case has shown that some stages can be partially borrowed and that every stage should be reliable, independent, and rapid. The reliability is achieved by decomposing every stage into independent building blocks encapsulating a specific elementary functionality. During execution, blocks providing an optional functionality could be skipped in a case of failure to continue processing without stopping a complete workflow. The independence of stages is ensured by decoupling stages one from

37

another; thus the workflow can be created by combining stages to fulfill specific requirements of an application case. Furthermore, decoupling allows to abstract from a particular view of analysis workflow and perform parallelization of the stages. High execution speed can be achieved by use of high-performance cluster-computing on GPUs or CPUs, exploiting hardware-specific optimizations, and an efficient implementation using proper programming languages. The data analy-
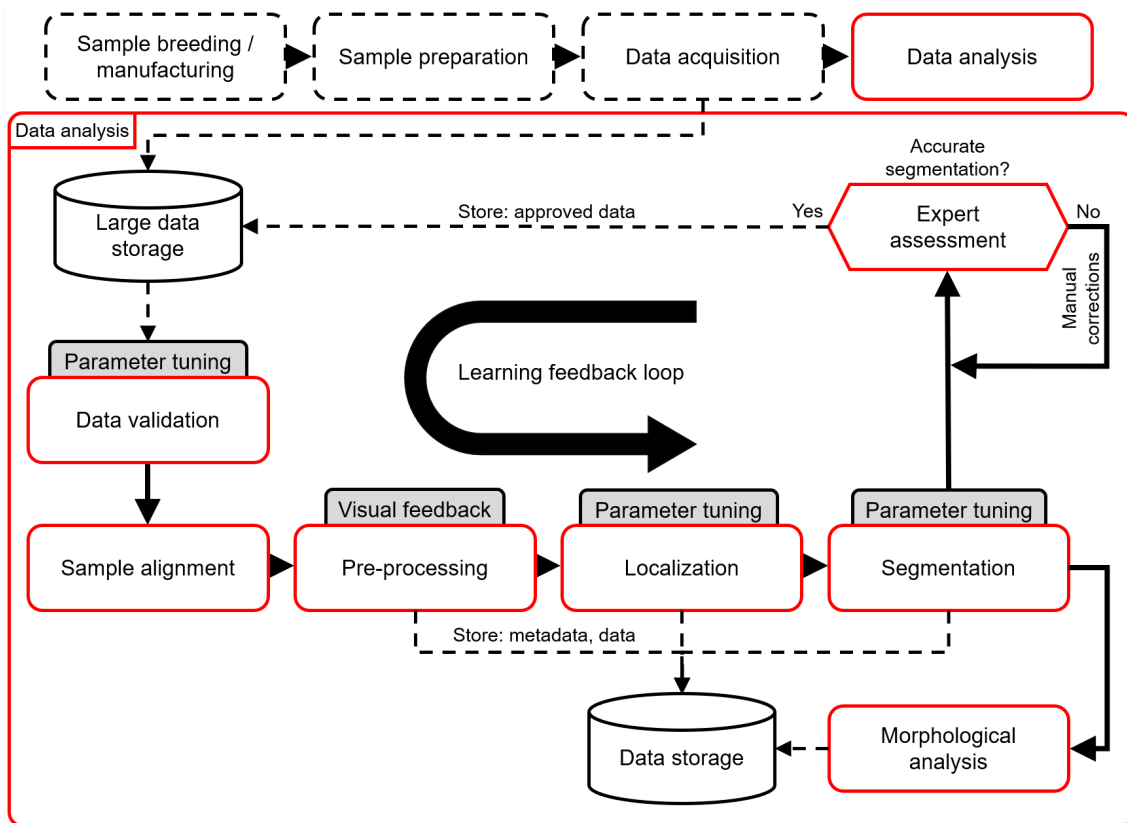
Figure 2.1.: The general diagram of the segmentation framework based on the custom data analysis workflow for µCT datasets (X-ray datasets) with the learning feedback loop, the visual feedback and parameter tuning components to find the optimal parameters of every processing stage. The red blocks represent new steps, stages or components in all the following figures.

sis workflow proposed in this thesis in Fig. 2.1 resembles the one from [70] regarding the functionality of some stages and architectural principles. It is composed of various independent processing stages as presented in Sec. 1.4.4 to analyze a series of similar samples. However, the order of stages is mainly determined by the nature of data and may vary depending on the application case. In the beginning, the data is validated to weed out the outliers, then the validated data should be spatially transformed to be co-aligned with some reference landmarks or another dataset. Then, the aligned data is transferred to the pre-processing stage to remove noise and artifacts from various sources inherent to the specific imaging setup. Afterward, the processed data is passed to the localization stage, where the structures of particular interest are localized and provided for the fol-

lowing stage, which extracts the localized structures and performs segmentation to delineate the target structures. At the final stage, all segmented structures are analyzed, and a set of morphological characteristics is estimated and stored. The obtained new segmentation is validated and stored into a dedication facility to improve the model at the next iteration of the learning feedback loop.

2.1.1. Learning feedback loop

This new concept is based on adding new segmented datasets to the storage, which allows to iteratively improve the prediction models for validation, localization, and segmentation in terms of accuracy of valid data recognition, bounding boxes detection, and structure delineation, respectively. Thus, the learning loop can be scheduled to perform retraining of models taking into account added datasets after a certain amount of time. During the retraining process, the workflow is not completely stopped, but rather all incoming processing jobs requiring some of the prediction models are added into a queue, where they are waiting until the training will be finished.

2.1.2. Expert assessment

Before passing a new segmented dataset into a dedicated storage, it should be first assessed by an expert in the domain. Since segmentation algorithms are usually trained at a limited number of datasets, the produced segmentation will always contain wrongly segmented structures, which could be barely noticeable or significantly distort the result. Therefore, it should be validated by the expert to avoid adding wrongly segmented data into a set of training data, which potentially could lead to lowering of segmentation accuracy. The expert can either correct wrongly segmented structures or reject dataset from adding to the training set.

2.1.3. Parameter tuning

The prediction models are based on deep-learning convolution neural networks, whose architecture completely depends on properties of datasets. At the moment, there is no theory explaining how to properly parameterize a specific model to get the best performance. Therefore, every prediction model during the retraining process performs a hyper-parameter optimization to derive the best-suited architecture for the datasets obtained during specific experiments. The hyper-parameter optimization is a cumbersome and lengthy process; therefore, it is performed only over a predefined set of parameters using the grid search approach.

2.1.4. Visual feedback

In some cases, data processing algorithms (*e.g.*, denoising) produce results of subjective nature, which can be only qualitatively evaluated. Therefore, visual feedback can help a user to find the best algorithm parameterization by providing a gallery of algorithm outputs obtained at various combinations of specified parameters.

2.1.5. Data storage

When a dataset is segmented and morphological analysis is performed, the produced statistics are stored in a specifically configured database. While, the new segmented dataset is then passed for validation to the expert, who check and correct segmentation if necessary and push it to the large data storage facility.

2.2. Data validation

The high-throughput experiments usually produce an enormous amount of data, which further is subjected to automated analysis due to impracticality of manual processing. The validation stage is required to exclude all data obtained in result of wrongly scanned samples to reduce loss in accuracy during further analysis. The incorrectly scanned samples could occur due to factors such as incorrect posi-
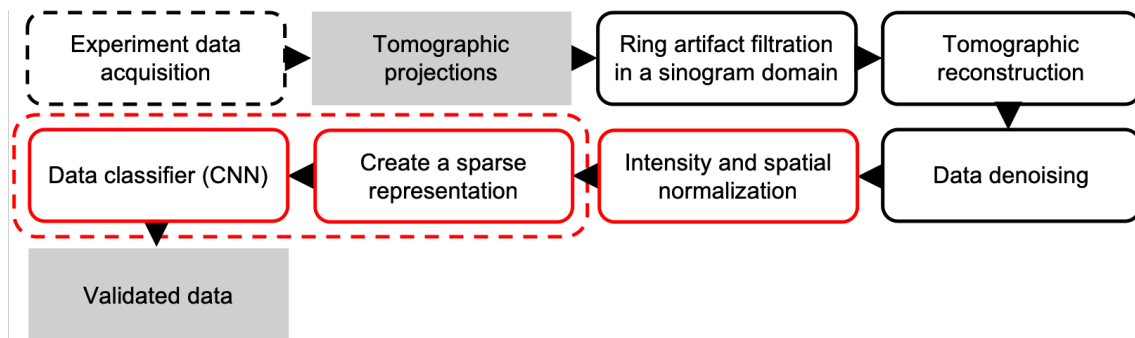
Figure 2.2.: The diagram of the data validation stage which allows detecting datasets of improperly scanned samples and excludes them from the further analysis steps. The red blocks represent new steps for data normalization, creating a sparse representation and training a data classifier.

tioning and alignment of the sample at the rotation stage (Fig. 2.3), an improperly selected sample preparation protocol, inaccurate experiment parameters, severe contamination with artifacts and other conditions which depend on the specific imaging technique and setup.
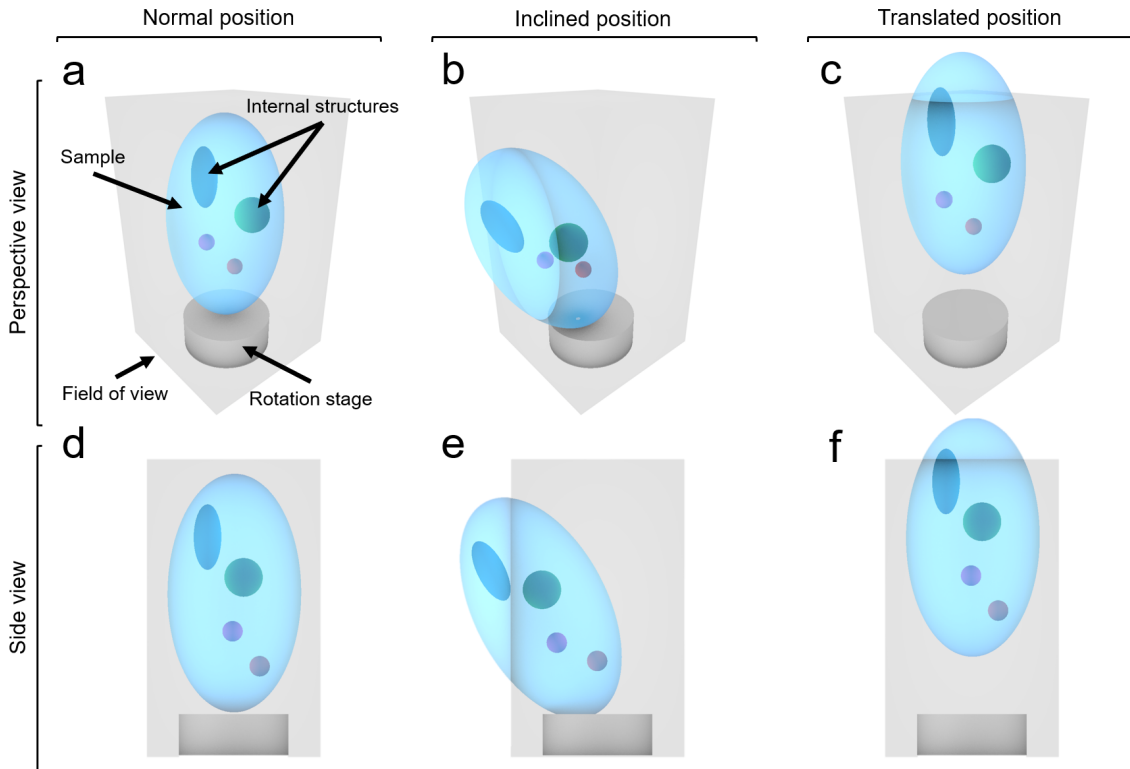
Figure 2.3.: The examples of the sample alignment to be handled in data validation: (a,d) normal position in the field of view; (b,e) inclined position, when the sample peeks out of the lateral sides of the field of view; (c,f) translated position, when the sample intersects the upper or bottom sides of the field of view.

The validation procedure can be formulated as a classification task, which is required to determine the state of the currently processed dataset. Similar tasks arise in various high-throughput studies using other imaging techniques such as bright-field microscopy to classify whole-body zebrafish deformations in multi-fish microwell plates [272, 273] and zebrafish embryos according to their defects [274–276] caused by various chemical substances. In the case of plant phenotyping, it is used to classify plant diseases and stresses employing ground and aerial platforms equipped with multiple sensors [277]. Also, it can be done manually, *e.g.*, to perform data quality control in phenotyping of mouse brain from structural μMRI datasets [278] or to estimate the quality of datasets coming from different mouse phenotyping centers [279]. Due to the variety of use cases, the strategy of validation procedure explicitly depends on the nature of data, *i.e.*, every application case has unique approach satisfying the specific requirements.

Here will be presented the algorithm for validation μCT datasets to exclude all datasets suffering from severe artifacts or where the sample is located out of the field of view. The projections acquired during X-ray experiments are represented as 2D grayscale images of the size equal to the detector size. Each pixel of projections has 16 or 32-bit color depth, which entirely depends on the detector properties. These projections are then subjected to the flat-field correction to remove hot

pixels of the detector, imperfections of the scintillator, and alleviate the instability of the X-ray beam. After that, they are transformed into a stack of cross-sections or so-called slices which allow revealing the internal structure of the sample with the help of tomographic reconstruction algorithms [12–14]. The reconstructed X-ray data could contain statistical noise, and various artifacts occurred due to the nature of X-ray imaging, such as ring and streak artifacts. These artifacts may severely contaminate the reconstructed data and make it unusable for further analysis; therefore, they should be suppressed as much as possible. The ring artifacts more frequently appear at the reconstructed data, since they depend on the detector and scintillator, which usually are not changed during the experiment. Therefore, the ring artifacts could appear on data of all scanned samples (Sec. 1.4.2). Whereas, the streak artifacts usually produced by samples containing highly absorbing parts surrounded by the weakly absorbing environment (Sec. 1.4.2).

Besides the artifacts and noise occurred due to the experiment design and the sample nature, the data could be spoiled by improper positioning of the sample as depicted in Fig. 2.3. It could happen if the sample was scanned incompletely, *i.e.*, a part of the sample is located out of the field of view. Such sample position violates its structural integrity, and the complete 3D dataset should be excluded from further consideration.

2.2.1. Tomographic reconstruction and artifacts filtration

The X-ray projections at first are converted into sinograms $\boldsymbol{I}_{sino}$ as described in Sec. 1.3 and proceed the ring artifact filtration $f_{ring}(\cdot)$ using the algorithm proposed in [47], which is relying on filtration of vertical lines in the wavelet domain. Every sinogram is wavelet-decomposed into the vertical, horizontal and diagonal details bands at different decomposition levels N_{wl}. Thus, the vertical and horizontal bands contain information about all vertical or horizontal stripes presented in the original image, correspondingly. After that, the bands carrying undesirable stripe information are Fourier transformed to compress the stripe information into narrow bands. Applying 1D Fourier transform to all columns of the vertical wavelet band will concentrate the stripe information around the X-axis of the produced Fourier spectrum. Then, these coefficients are multiplied with a Gaussian function of σ_{dmp} to remove those located near to the X-axis. Next, the coefficients are transformed back to the wavelet domain with 1D inverse Fourier transform, and the de-striped sinogram $\boldsymbol{I}'_{sino}$ is obtained with the inverse wavelet transform applied to the refined wavelet coefficients.

$$\boldsymbol{I}'_{sino} = f_{ring}(\boldsymbol{I}_{sino}; N_{wl}, \sigma_{dmp}) \tag{2.1}$$

Finally, applying tomographic reconstruction algorithms to the de-striped sinograms (Eq. (2.1)) produces the reconstructed slices forming the dataset $\boldsymbol{I}_{reco}$ with suppressed ring artifacts as shown at Fig. 2.4b.
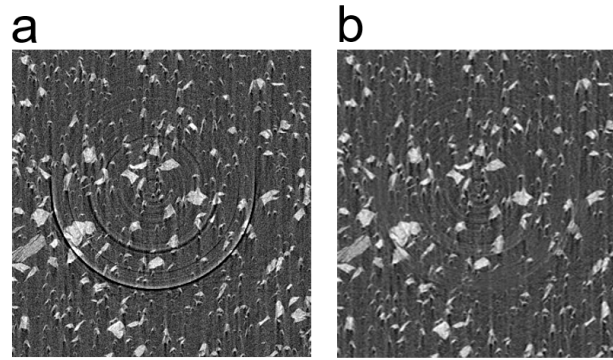
Figure 2.4.: The example of applying ring artifact correction to the real dataset: before (a) and after (b) correction.

2.2.2. Data denoising

The correction of ring artifacts can help either to remove most of them or at least significantly decrease their severity. However, the reconstructed data may still suffer from the statistical noise, which was not filtered out with ring artifacts (Fig. 2.5a).

At the validation stage, the data is preliminarily checked for suitability to the further analysis. Thus, it is not reasonable to use computationally expensive non-linear 3D filtering operations. Therefore, the 2D median filtering $f_{median}(\cdot)$ with the window of the specified size W_{med} is applied to all slices along each orthogonal axis ax to suppress the statistical noise in the reconstructed data

$$\boldsymbol{I}'^{ax}_{reco} = f_{median}(\boldsymbol{I}^{ax}_{reco}; W_{med}). \tag{2.2}$$

This results in the denoised data $\boldsymbol{I}'_{reco}$, and it allows partially mitigate weak streak artifacts and homogenize internals of structures that will benefit in following steps (Fig. 2.5b). However, the size of W_{med} should be selected not larger than the size of the smallest structure of interest.
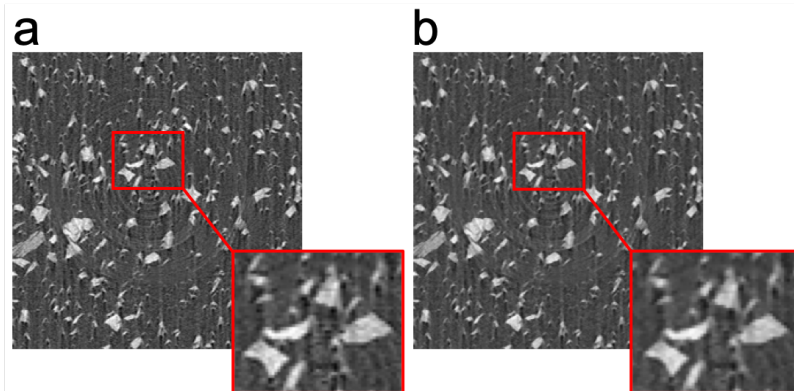

Figure 2.5.: The example of applying denoising process to the real dataset: before (a) and after (b) denoising.

43

2.2.3. Intensity and spatial normalization

To proceed with further steps, the 32-bit float data should be converted into the 8-bit unsigned integer data using a histogram stretching and subsequent discretization denoted by $f_{norm}(\cdot)$. The range of intensities in the 32-bit data is globally selected by specifying two percentiles (P_{low}, P_{high}) to define an intensity range for conversion to 8-bit, which allows decreasing the total size of the dataset by factor of four. The specified percentiles are used to determine a global intensity range for the dataset by averaging intensity ranges obtained at equally-spaced slices at each N_{qss} percents along the selected orthogonal axis. The careful selection of these percentiles is a vital step because it defines the further intensities of the dataset after conversion:

$$\boldsymbol{I}_{norm} = f_{norm}(\boldsymbol{I}'_{reco}; P_{low}, P_{high}, N_{qss}). \tag{2.3}$$

The selected percentiles usually can be reused for data obtained from samples of similar nature, prepared by the same protocol, and acquired using the same imaging setup and experiment conditions. In a case of different datasets, unique percentiles should be selected individually for each dataset

The size of reconstructed datasets may routinely reach dozens of gigabytes due to high-resolution detectors. Unfortunately, such large sizes make troubles for the validation stage, since it slows down the analysis process and increases memory consumption. To avoid this, the dataset should be scaled:

$$\boldsymbol{I}_{scaled} = f_{scale}(\boldsymbol{I}_{norm}; S_{val}) \tag{2.4}$$

to fit the predefined size S_{val}, thereby spatially unifying datasets obtained with detectors of different resolutions. Moreover, adjustment of the size S_{val} allows significantly decrease the total size of the scaled dataset.

2.2.4. Data classifier

The previous attempts in validation tasks utilizing CNNs have shown high efficiency and accuracy [272, 273]. However, they all were developed mainly for 2D images, and hence cannot be directly applied to 3D data, since the network should take into account the third dimension. Therefore, the CNN was modified for a 3D case, allowed to parameterize the architecture and adapt for a sparse representation of 3D data to reduce a GPU memory burden.

The sparse representation

The configuration of network $CNN(\cdot)$ imposes requirements on dimensionality and size of the input data. Particularly, in the case of CNN, the input data size is extremely restricted by the amount of available GPU memory, because it directly

influences on the network architecture, and subsequently on the number of trainable weights. On the other hand, the dataset must have enough slices along the third dimension to maintain the structural integrity of the scanned sample. Thus, an optimal sparse representation is produced by $f_{sparse}(\cdot)$ for each dataset scaled to the size of S_{val} by keeping only equally-spaced slices along the selected axis ax with the step of N_{qss} percents:

$$\boldsymbol{I}_{sparse}^{ax} = f_{sparse}(\boldsymbol{I}_{scaled}^{ax}; N_{qss}). \tag{2.5}$$

The creation process of a sparse representation of a phantom dataset along a sagittal view is illustrated in Fig. 2.6, such representations can be produced for each orthogonal axis if several networks are employed in the validation process.
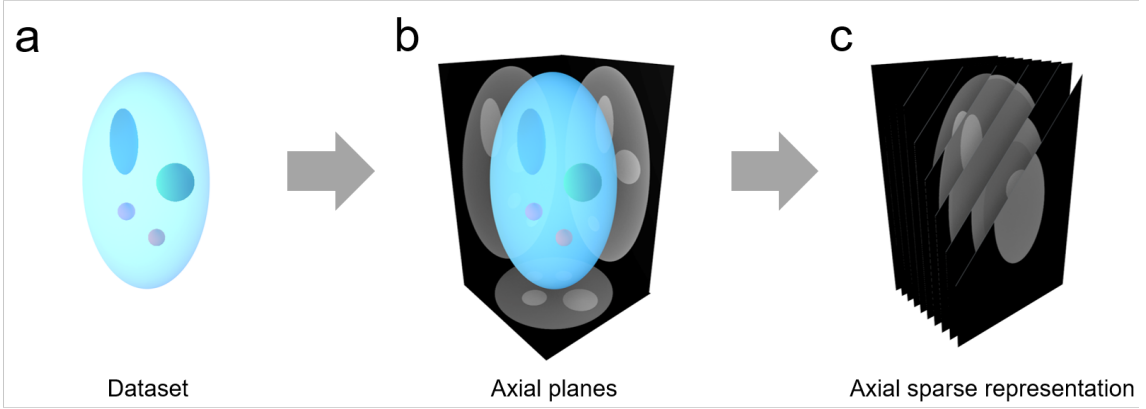
Figure 2.6.: The schematic illustration of the creation process of a sparse representation of a phantom dataset: (a) a phantom dataset; (b) slices obtained along each orthogonal axis; (c) a sparse representation along the Y-axis.

For the training phase, every sparse representation should be accompanied by an integer class label C. The sparse representations obtained from different axial planes of the same dataset are assigned the same class label. For example, the sample alignments presented in Fig. 2.3(a, d) and Fig. 2.3(b,e) can be denoted by zero- and one-class labels, correspondingly. However, they are not understandable for the CNN due to the inability to estimate a loss function for them. Therefore, integer class labels should be converted into a suitable view for the CNN by means of a one-hot encoding algorithm $f_{ohe}(\cdot)$, which encodes integers by unique sequences of binary codes:

$$\begin{aligned} \boldsymbol{C}' &= f_{ohe}(C) \\ \boldsymbol{C}' &= [e_0, e_1, \dots, e_{N_c-1}] \\ e_i &\in \{0, 1\} \\ i &\in \{0, 1, \dots, N_c - 1\}. \end{aligned} \tag{2.6}$$

The network architecture

The type of network mentioned in [272] was extended to handle 3D data and generalized through parameterization of the architecture: the network depth N_d or the minimal size of the receptive field at the last convolution layer S_{mrf}, the number of filters at the first convolution layer N_{bf}, the number of neurons at the fully-connected layer N_{fc} and the amount of dropout N_{dp}. These parameters allow performing a hyper-parameter optimization to find the best suitable network architecture for the specific application case.

The architecture of a network is presented in Fig. 2.7, and consists of several types of layers, which are placed in the following order: an input layer, several blocks of two convolutional layers and following a max-pooling layer, a fully-connected layer, and an output layer. This order of layers is inherent to the networks aimed at prediction of data classes expressed via scalar values.

The data of size $W \times H \times S$, where W, H, and S are the width, the height and the number of slices along a sparse dimension, is passed to the input layer of equal size S_{val}. Then, the data is propagated through a network of length N_d, starting from a sequence of convolution layers of size $W^{d-1} \times H^{d-1} \times S$ formed by convolving the previous layer with a stack of $N_{bf} \cdot d$ filters of size $3 \times 3 \times S^{d-1}$ and passed through a ReLU function to produce the activations of the current layer at the network level d. After passing convolutional levels, the data is transferred to the max-pooling layer producing the data of size $W^{d-1}/2 \times H^{d-1}/2 \times S$ by performing a max operation over $N_{bf} \cdot d$ feature maps from the previous level with a filter of size $2 \times 2 \times 1$. When the propagated data reaches the fully-connected layer at N_d level consisting of N_{fc} so-called neurons with a ReLU activation function, it is linearized and mapped to all neurons. These neurons are passed to the dropout layer [280], which randomly zeroes a fraction N_{dp} of the neurons for a regularization purpose. Finally, the rest of the neurons are transferred to the output layer, where the activations are obtained with a sigmoid function, and the errors with respect to the ground truth are estimated, which are then back-propagated through the network to adjust the weights.
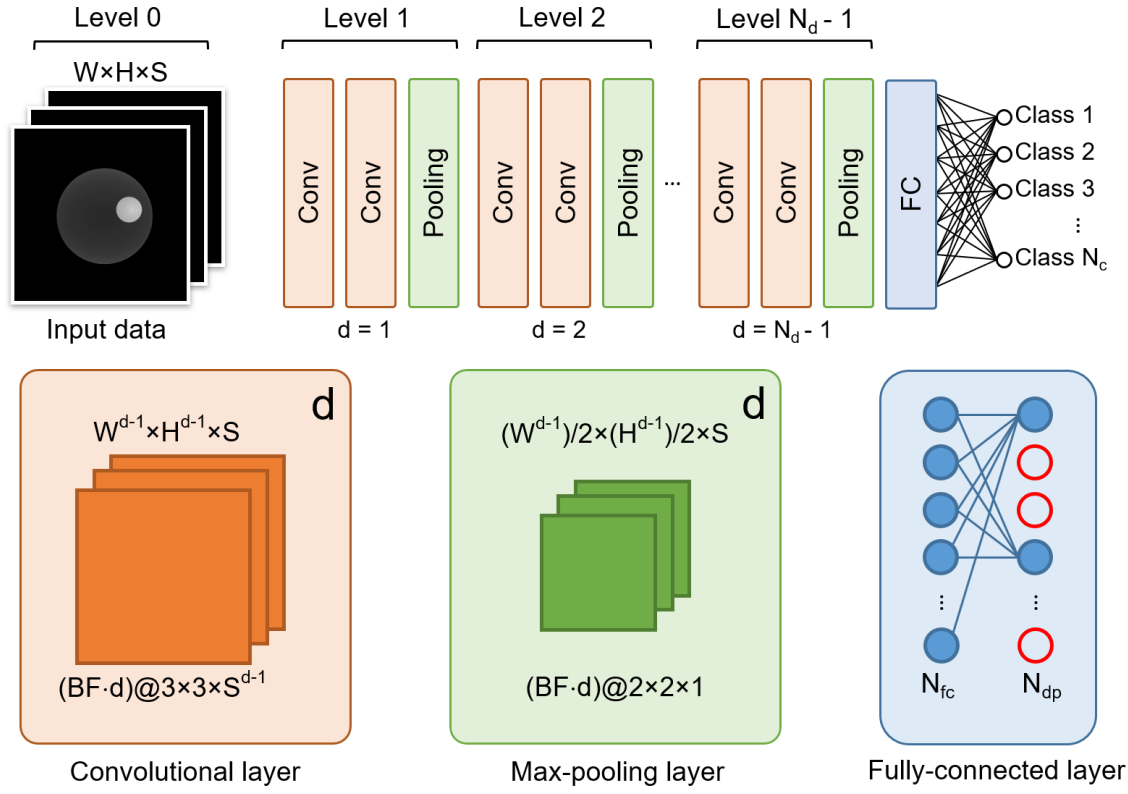
Figure 2.7.: The architecture diagram of the CNN involved in the validation process. The lower row illustrates a convolutional and a max-pooling layer for level d, and a fully-connected layer.

The network performs an optimization procedure to minimize the output error by minimizing a specific loss function, *e.g.*, in the case of validation it is a cross-entropy:

$$f_{loss}(\boldsymbol{y}) = -\frac{1}{N_s} \sum_{i=0}^{N_s-1} \sum_{c=0}^{N_c-1} \mathbf{1}_{y_i \in C_c} \log p(y_i | C_c), \qquad (2.7)$$

where N_s is the number of observations, N_c is the number of classes or categories, C is a set of classes, and $\mathbf{1}_{y_i \in C_c}$ is the indicator function of the i-th observation belonging to the c-th category. The $p(y_i | C_c)$ is the probability predicted by the network for the i-th observation to belong to the c-th category. At each pass, the network outputs a vector of $\boldsymbol{P}$ probabilities, each describing the probability that the network input should be classified as belonging to the respective class.

The prediction process

The parameterized CNN can be expressed as Eq. (2.8), which accepts a sparse representation of a dataset $\boldsymbol{I}_{val}^{ax}$ along axis ax of size S_{val}, and produces a prediction vector $\boldsymbol{P}^{ax}$. This vector contains probabilities representing belonging of the

input data to the specific class:

$$\boldsymbol{P}^{ax} = \text{CNN}^{ax}(\boldsymbol{I}^{ax}_{val}; S_{val}, N_{bf}, S_{mrf}, N_{fc}, N_{dp}). \tag{2.8}$$

The probabilities are thresholded $f_{th}(\cdot)$ with the specified threshold value γ_{th} to obtain the class encodings:

$$\begin{aligned} \boldsymbol{C}^{ax} &= f_{th}(\boldsymbol{P}^{ax}; \gamma_{th}) \\ \boldsymbol{C}^{ax} &= [e_0, e_1, \ldots, e_{N_c-1}] \\ e_i &\in \{0, 1\} \\ i &\in \{0, 1, \ldots, N_c - 1\}, \end{aligned} \tag{2.9}$$

where $\boldsymbol{C}^{ax}$ is a one-hot encoded vector of classes.

The strategy of data augmentation

Data augmentation allows producing synthetic data based on real data to increase a variety of sample appearance in a population. As was described in Sec. 1.4.3, due to the nature of CNNs, their full potential can be leveraged only if there is enough of training data. However, in some cases, experiments may be expensive, time consumable or can be performed only a limited number of times that significantly reduces the number of produced datasets which could participate in a training.

When datasets from experiments are limited, the acquired datasets of similar samples with analogous imaging setup can be reused to generate additional datasets. The process of generation consists in applying a set of randomly or manually parameterized geometrical (Fig. 2.8) and intensity transformations to the available datasets. The combination of such transformations is able to produce the new dataset with plausible spatial and intensity modifications controlled by a set of parameters. It allows teaching the network about possible variations in pose and morphology of the sample and improves the robustness and generalization of the final prediction model.

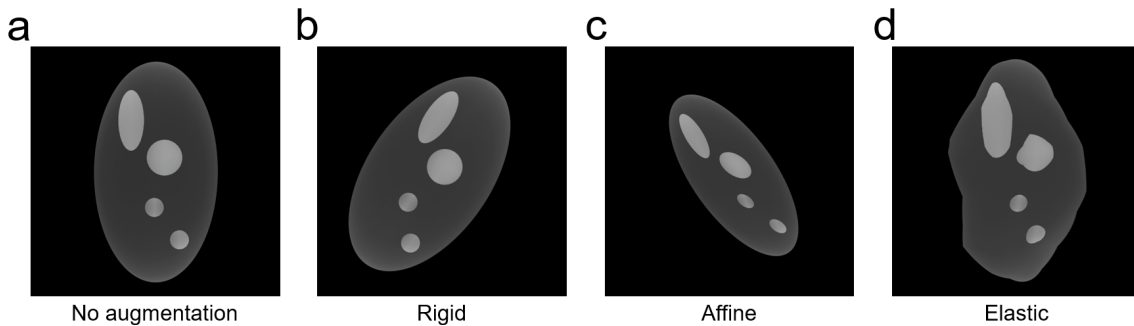a   b   c   d

No augmentation   Rigid   Affine   Elastic

Figure 2.8.: The results of applying geometrical transformations involved into data augmentation process: (a) no transformations; (b) rigid transformation; (b) affine transformation; (c) piece-wise affine transformation.

The geometrical transformations differ one from another by complexity they can provide. As for example, rigid transformation allows only for translation and rotation, whereas affine transformation includes both of them and additionally scaling and shearing operations [281]. However, they perform the global transformation of the coordinate system of the dataset to another coordinate system, thereby not allowing for local transformations to deform various parts of the dataset independently. The piece-wise affine transformation can be used to perform local deformations by placing a regular grid on the dataset and randomly move the neighborhood of these points along sampled force vectors with affine transformations.

Thus, additional datasets can be produced by an accurately parameterized augmentation process to generate datasets that are comparable to the real ones. It can improve the prediction accuracy in the case of an insufficient amount of data representing positive or negative cases for the training.

2.3. Sample alignment

The goal of alignment or registration is to spatially align the moving image $I_M(x)$ to the fixed image $I_F(x)$, where the moving image is the original image, which undergoes various spatial transformations. This is a procedure of finding a coordinate transformation $T(x)$ that makes the transformed moving image $I_M(T(x))$ be aligned with $I_F(x)$ as schematically illustrated in Fig. 2.9. The moving image transformation can be described as:

$$I_M(T(x)) = f_{mv}(I_M, x, T(x)), \qquad (2.10)$$

where f_{mv} is a mapping function of the image I_M with the coordinates x to the transformed coordinates $T(x)$. The alignment quality of these images is estimated with a cost function $\mathcal{C}(T; I_F, I_M)$ which is minimized if the images are aligned. Thus, an optimization process tries to minimize a cost function to find the optimal transformation parameters.
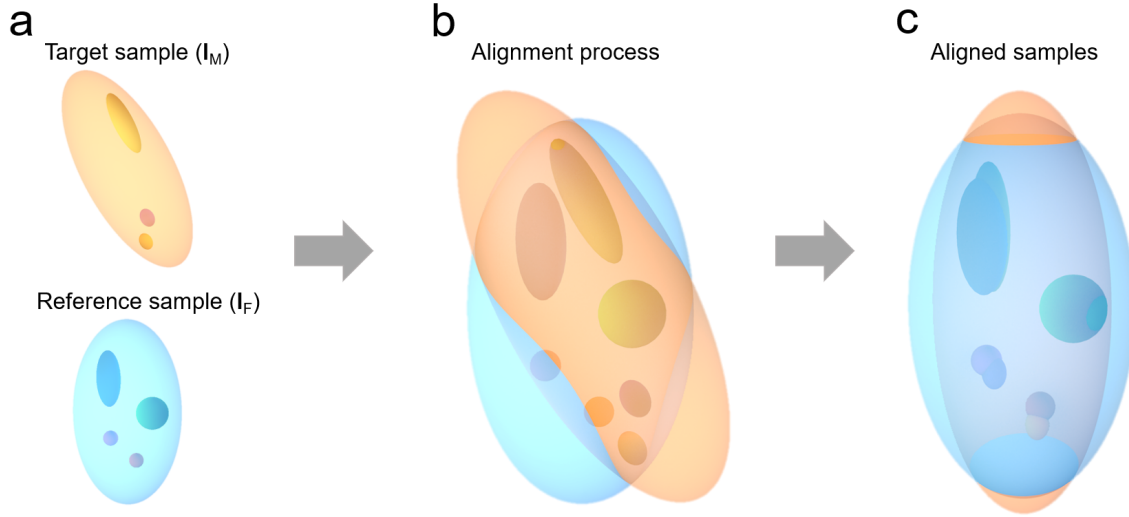
Figure 2.9.: The schematic representation of the alignment process, where the target ($\boldsymbol{I}_M$) sample (a) is passing through a series geometrical transformations (b) to find one, which will align it to the reference ($\boldsymbol{I}_F$) sample (c).

Usually, the registration task is accompanied by a number of choices to be made, such as the optimization method [282, 283], the cost function [284–289], the model of coordinate transformation [287, 290], the interpolation algorithm to correctly transform the moving image [291], the sampling and multi-resolution strategies [292].

2.3.1. Overview

The registration problem can be formulated as an optimization problem, where the selected metric defined by the cost function $\mathcal{C}$ is minimized with respect to parametrized transformation $\boldsymbol{T}$. Since, the transformation is defined by a number of parameters, only a limited number of transformations is possible. The final optimization problem can be written as:

$$\boldsymbol{\mu} = \underset{\boldsymbol{\mu}}{\operatorname{argmin}}\, \mathcal{C}(\boldsymbol{T_\mu}; \boldsymbol{I}_F, \boldsymbol{I}_M) \tag{2.11}$$

where $\boldsymbol{\mu}$ is a vector carrying the transformation parameters, $\boldsymbol{T_\mu}$ denotes the transformation $\boldsymbol{T}$ parametrized by $\boldsymbol{\mu}$. This minimization problem Eq. (2.11) is solved with an iterative optimization method in a multi-resolution framework. The schematic representation of the entire process is shown in Fig. 2.10, which is the adaptation of the scheme presented in [293].
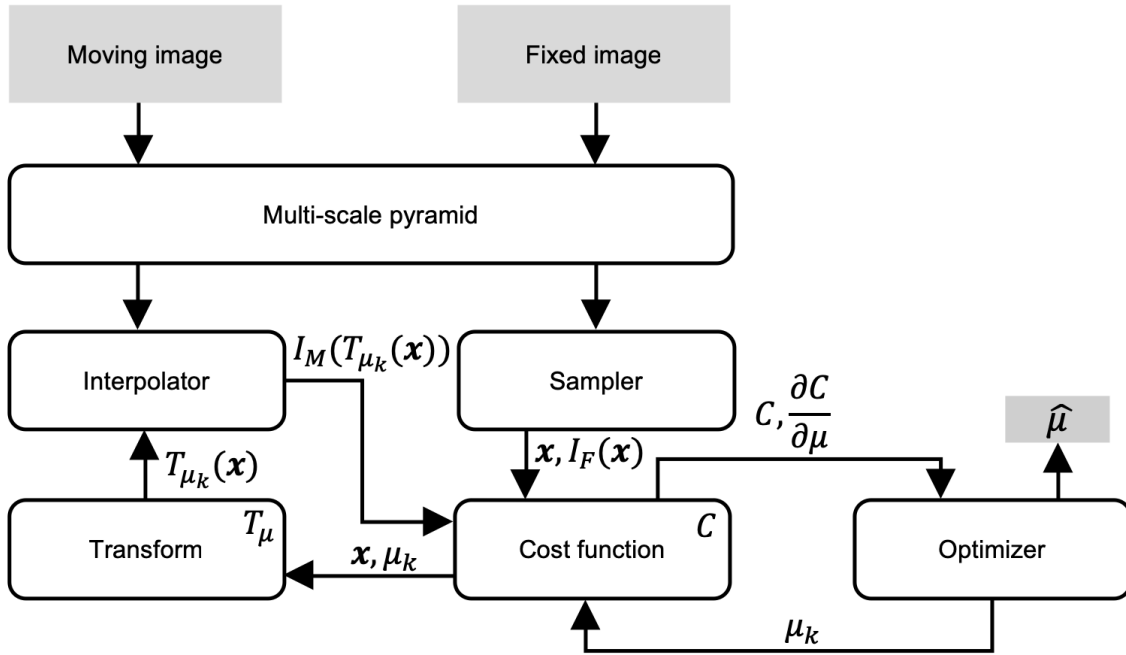
Figure 2.10.: The diagram of the alignment stage describing the general mechanisms to align a target (moving) dataset to a reference (fixed) one.

2.3.2. Cost function

The cost function $\mathcal{C}$ defines a metric to estimate the similarity between the fixed image $\boldsymbol{I}_F(\boldsymbol{x})$ and the transformed moving image $\boldsymbol{I}_M(\boldsymbol{T}(\boldsymbol{x}))$. There is a lot of metrics to use, such as mean square difference (MSD) [284], normalized correlation (NC) [285, 286], mutual information (MI) [287, 288, 294], normalized MI (NMI) [289], α-MI [295, 296], and the κ-statistic [297]. For an instance, the MI metric is defined as:

$$
\begin{aligned}
\mathcal{MI}(\boldsymbol{X}, \boldsymbol{Y}) &= H(\boldsymbol{X}) + H(\boldsymbol{Y}) - H(\boldsymbol{X}, \boldsymbol{Y}) \\
&= \sum_i \sum_j p_{\boldsymbol{xy}}(i, j) \log_2 \frac{p_{\boldsymbol{xy}}(i, j)}{p_{\boldsymbol{x}}(i)} p_{\boldsymbol{y}}(j),
\end{aligned}
\tag{2.12}
$$

where $\boldsymbol{X}$ and $\boldsymbol{Y}$ are fixed and moving images, $i \in \boldsymbol{X}$ and $j \in \boldsymbol{Y}$ are values, $p_{\boldsymbol{x}}(i)$ and $p_{\boldsymbol{y}}(j)$ are the probabilities of occurrence of these values in the corresponding images, $p_{\boldsymbol{xy}}(i, j)$ is the joint probability of these values occurring together, H is the Shannon entropy, and $H(\boldsymbol{X}, \boldsymbol{Y})$ is a joint entropy. The voxels are sampled from the domain according to the selected sample strategy which will be described later.

The MSD metric is suitable only for images with similar intensities, usually, these images are from the same modality or obtained with the same parameters of the imaging setup. In contrast to MSD, the NC metric allows being used when intensities of the fixed and moving images have an affine relation. The most relaxed

conditions are provided by the MI, NMI and α-MI metrics, which assume that there is only a statistical relation between intensities of the images. Therefore, these metrics can be used for both intra- and inter-modal registration tasks. The κ-statistic metric is developed for aligning of binary images by measuring overlap of segmented regions.

2.3.3. Geometric transformations

The severity of deformation is defined by the parametrization of the coordinate transformation model T_μ. The example of the affine transformation model allowing for translation, rotation, scaling and skew of the images:

$$T_\mu(x) = Ax + t, \tag{2.13}$$

where A is a matrix incorporating rotation, scaling and skew, and t denotes the translation vector. The parameter vector μ consists of the matrix elements a_{ij} and the components of the translation vector:

$$\mu = \left(a_{11}, a_{12}, a_{13}, a_{21}, a_{22}, a_{23}, a_{31}, a_{32}, a_{33}, t_x, t_y, t_z \right)^T. \tag{2.14}$$

The degrees-of-freedom of the deformation is completely determined by the parametrization vector μ, so for the rigid transformation model consisting of translation and rotation, only the elements of the A matrix responsible for rotation and the translation vector should be specified.

2.3.4. Optimization process

The optimization problem Eq. (2.11) can be solved as a result of the optimization procedure. At each iteration k, the vector of the current transformation parameters μ_k is updated by taking a step in the direction d_k to improve the current solution:

$$\mu_{k+1} = \mu_k - a_k d_k, \tag{2.15}$$

where a_k is a value defining the step size. A lot of optimization methods use similar update schemes [283], such as gradient descent, quasi-Newton, nonlinear conjugate gradient, evolution strategy, and stochastic gradient descent methods. However, each of them provides own definition for a_k and d_k. In the case of a gradient descent method, the direction of search is defined as the derivative of the cost function $\partial\mathcal{C}/\partial\mu$ evaluated at the current position μ_k.

2.3.5. Sampling strategies

The estimation of the cost function $\mathcal{C}$ and its derivative $\partial\mathcal{C}/\partial\mu$ requires to select a set of voxels $\boldsymbol{x} \in \boldsymbol{X}$ from the fixed image domain as stated in Eq. (2.12). The easiest and straightforward way is to use all voxels from the fixed image, whose downside is a significant computational burden in a case of large images. However, the more widespread approach implies that a set of voxels is selected randomly or from a uniform grid defined over the fixed image. In another strategy, voxels are picked only from remarkable image features like edges [298]. Also, the approach randomly samples voxels at non-voxel locations has shown to improve the smoothness of the cost function [299, 300].

All sampling strategies may be supplied with statically or dynamically defined image masks over the image $\boldsymbol{X}$. The dynamic masks are randomly defined at each iteration of the optimization procedure. The static masks are specified once automatically or manually and do not change during the iterative process. The mask forces the sampling algorithm to pick points only from the specified ROI, which in some cases allows for increasing the quality of alignment.

2.3.6. Interpolation

Calculation of the cost function involves evaluation of $\boldsymbol{I}_M(\boldsymbol{T}_\mu(\boldsymbol{x}))$ at non-voxel locations. Thus, the interpolation is required to correctly estimate this transformation of the moving image. The existing interpolation methods such as nearest-neighbor, linear and B-spline algorithms vary in quality and speed [301, 302], and therefore should be selected with a trade-off in mind. As for the fixed image, it does not require any interpolation with the most sampling strategies, because the image is sampled at the voxel positions.

2.3.7. Hierarchical approach

The multi-resolution strategies [292] are the important part of the image registration process. These strategies allow for a gradual increase of the transformation model complexity when approaching from a coarse resolution level to a finer one to introduce more fine-scale transformations. Moreover, the parameters of the selected sampling strategy and the cost function may be subjected to a hierarchical strategy to adaptively change the behavior on approaching to finer levels. As an example, in [294] was suggested to gradually increase the number of histogram bins for the estimation of the MI and NMI metrics.

2.4. Data pre-processing and normalization

The goal of this stage is to suppress the statistical noise and isolate the object of interest from excessive surrounding empty space and unimportant structures and prepare data for further hierarchical analysis. This stage is widely used in analysis of medical CT datasets as a part of the segmentation workflow for abdominal organs of patients; in plant phenotyping to prepare photographs for identification and classification; to process fluorescent images in zebrafish phenotyping. The data pre-processing became an essential part of any data analysis workflow because raw data from a detector system usually suffer from various kinds of noise and artifacts, which should be suppressed or removed before to proceed with the further analysis.
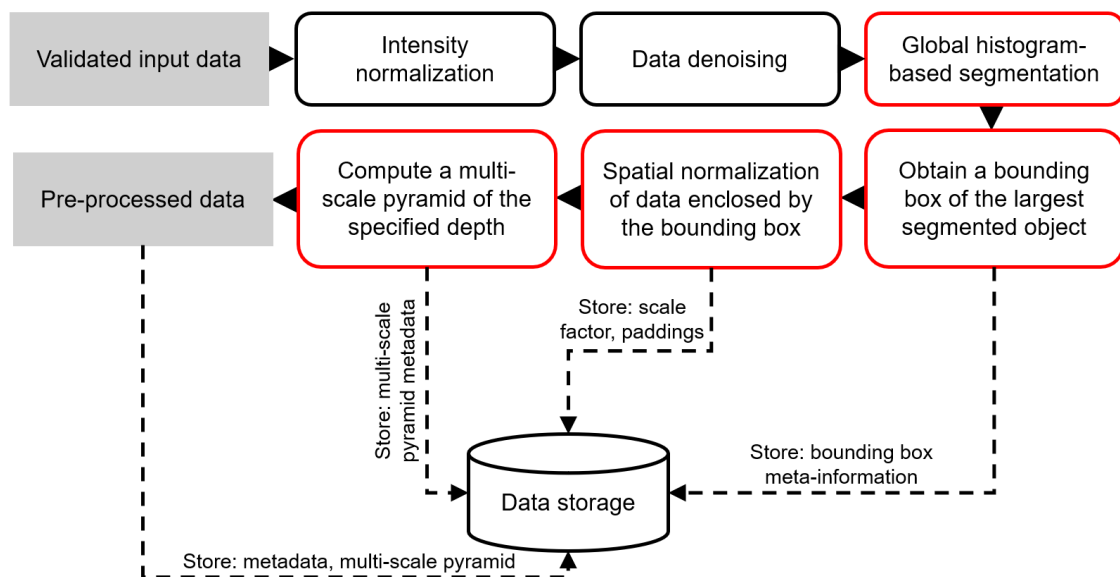
Figure 2.11.: The diagram of the pre-processing stage which prepares data for the further processing steps and analysis. The red blocks represent new steps for rough data segmentation, extraction of a segmented structure, spatial data normalization, and creation of a multi-level scale pyramid.

The proposed pre-processing pipeline (Fig. 2.11) is partially based on early studies [303–307] aimed at segmentation and analysis of data produced at various X-ray imaging devices, such as medical CT scanners, X-ray laboratory sources, and synchrotron light sources.

In the previously presented workflows, the input data at first proceeds the intensity normalization to convert gray values to some intensity range. In this work, this intensity range is defined as an average of intensity ranges estimated at equally-spaced slices of the µCT data along the specified orthogonal axis and step, as previously described in Eq. (2.3).

The converted data is then processed with an anisotropic filter to suppress the statistical noise. There are a lot of denoising algorithms potentially applicable to X-ray data. However, the non-local means algorithm $f_{nlm}(\cdot)$ [308] is the most cost-effective and not requiring any *a priori* knowledge and training procedures while ensuring the best quality among other denoising approaches [309] and demonstrating efficiency for denoising of tomographic data. The algorithm estimates every pixel as the weighted average of all gray values within the search window W_{sw}. The weights indicate the degree of similarity between a patch of size W_{tp} centered at the estimated pixel and all patches within the search window centered at the respective pixels. The smoothing parameter γ_{sm} controls the extent of averaging. The high values of γ_{sm} promote removing noise and at the same time, structural details, whereas low values preserve details, but as well noise. The denoising algorithm is applied along any specified orthogonal axis ax of the dataset $\boldsymbol{I}_{norm}^{ax}$, and can be expressed as:

$$\boldsymbol{I}_{norm}^{'ax} = f_{nlm}(\boldsymbol{I}_{norm}^{ax}; \gamma_{sm}, W_{tp}, W_{sw}). \tag{2.16}$$

Then, the dataset with suppressed noise $\boldsymbol{I}_{norm}^{'ax}$ is subjected to edge extraction procedure (*e.g.*, the Sobel operator $f_{sobel}(\cdot)$) to extract the region containing the object of interest:

$$\boldsymbol{I}_{edge} = f_{sobel}(\boldsymbol{I}_{norm}^{ax}). \tag{2.17}$$

Afterward, the global thresholding $f_{gth}(\cdot)$ is applied to the data with enhanced edges $\boldsymbol{I}_{edge}$, which results in the binarized contours of structures:

$$\boldsymbol{I}_{th} = f_{gth}(\boldsymbol{I}_{edge}). \tag{2.18}$$

The sequence of morphological operations $f_{morph}(\cdot)$ such as erosion, dilation, closing, opening and fill holes is applied to obtain the closed segmentation of the object of interest. These operations are parameterized by the size of the structural element S_{se} and the number of iterations N_{mi}, whose values depend entirely on the heuristic knowledge about the imaging setup and the nature of data. Thus, applying this sequence to the data leads to the fully closed binary mask:

$$\boldsymbol{I}_{th}^{'} = f_{morph}(\boldsymbol{I}_{th}; S_{se}, N_{mi}). \tag{2.19}$$

In the following step, the connected-components analysis $f_{cca}(\cdot)$ is performed to determine the largest bounding box B^* and isolate the object of interest:

$$\begin{aligned} \boldsymbol{B} &= f_{cca}(\boldsymbol{I}_{th}^{'}) \\ \boldsymbol{B} &= [B_0, B_1, \ldots, B_{M-1}] \\ B_i &= [x_{min}, x_{max}, y_{min}, y_{max}, z_{min}, z_{max}] \\ i &\in \{0, 1, \ldots, M-1\} \\ B^* &= \max(\boldsymbol{B}) \text{ by volume}, \end{aligned} \tag{2.20}$$

where x, y and z are coordinates along the corresponding orthogonal axes, and operator $\max(\cdot)$ determines the bounding box enclosing the largest volume. The

data contained by a bounding box $\boldsymbol{I}'^B_{th}$ is then zero-padded and scaled $f_{snorm}(\cdot)$ to fit the specified size S_{norm} to normalize dataset in terms of size to other datasets participating in the analysis:

$$\boldsymbol{I}^B_{sn} = f_{snorm}(\boldsymbol{I}'^B_{th}; S_{norm}). \tag{2.21}$$

Finally, the rescaled and zero-padded dataset $\boldsymbol{I}^B_{sn}$ is used to create the multi-scale pyramid $f_{pyr}(\cdot)$ to make possible a hierarchical analysis:

$$\begin{aligned} \boldsymbol{H} &= f_{pyr}(\boldsymbol{I}^B_{sn}; N_L, \gamma_{ds}, \sigma_{pyr}) \\ \boldsymbol{H} &= [\boldsymbol{I}_0, \boldsymbol{I}_1, \ldots, \boldsymbol{I}_{N_L-1}], \end{aligned} \tag{2.22}$$

where N_L is the number of layers for the pyramid. Every next level is produced by downscaling the dataset from the previous layer by the factor of γ_{ds} and subsequent smoothing with the Gaussian filter of σ_{pyr}. The multi-scale pyramid enables to analyze the objects of interest at various scale levels providing a different degree of details. The initial scale level of pyramid H_0 is created by applying all previous pre-processing steps to the input dataset. The total processing time can be dramatically reduced by performing time-consumable operations at the coarsest possible level. The obtained results further can be remapped to higher or lower levels. The algorithm for computing of the multi-scale pyramid is described in [98].

All meta-information about the bounding boxes, the scale factors, the number of added zeros and the parameters of the multi-scale pyramid is stored for further use.

Thresholding algorithm selection

The thresholding segmentation methods rely on the detection of peaks and valleys in a histogram and are very sensitive to the content of the data. Due to a vast variety of content, there were proposed a lot of approaches [310] for analyzing histograms, because typically, each case requires a unique approach.

In early systematic studies on thresholding segmentation methods [311, 310], there were attempts to formulate a methodology for selection of segmentation method for non-destructive testing images such as thermal, ultrasonic, light microscope, ceramic, cloth images and synthesized images. In these studies, each method was evaluated for each image from the specific application case.

However, despite that various methods were evaluated, the obtained results can only be considered as a recommendation, because the number of images in each case was insufficient, and therefore hardly can be used as an established methodology. Moreover, due to probable variations in the content of the images, even the recommended methods could fail to determine a proper thresholding value. Among the studied methods, no one was systematically evaluated on X-ray data,

and therefore there are no recommendations for the method selection in this case.

Based on heuristic knowledge, in this work, it is proposed to use the entropy-based methods when the fraction of data subjected to segmentation is small (*e.g.*, small particles embedded in a polymer scaffold). The clustering-based methods are more suitable if the object of interest occupies the large fraction of data, and the histogram has emphasized peaks. The determination of the proper method is possible by visual inspection of results produced by each method on the representative slice of X-ray data.

2.5. Localization of internal structures

In the previous chapter, it was outlined that with the development of deep learning methods [62], the organ localization approaches based on CNNs started to emerge and overcome the traditional ones, which mainly relied on classical machine learning methods like SVM, Linear regression, Decision Forests, and others. These classical methods required to design features manually, which is cumbersome due to restrictions of human perception, whereas the deep learning methods can automatically extract hierarchical features in domains invisible for the naked human eye.

2.5.1. Overview

In this section, the 3D localization method proposed by Humprie-Mamani *et al.* [161] for medical CT datasets will be described, generalized and adopted in the context of high-throughput µCT experiments. This approach is based on the detection of structures presented in 2D slices of X-ray data, whose schematic overview is illustrated in Fig. 2.12.
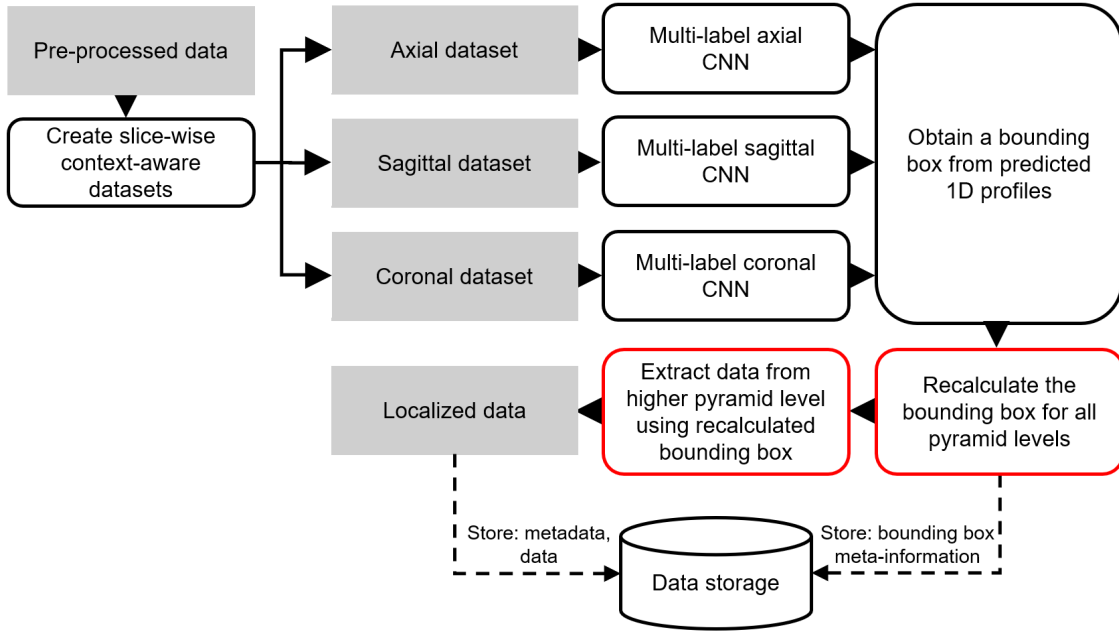
Figure 2.12.: The diagram of a general view of the localization stage, which allows determining bounding boxes of target structures in a dataset. The red blocks represent new steps for recalculation of obtained bounding boxes for pyramid levels and extraction of target structures using these bounding boxes.

The method relies on three multi-label CNNs, one for each orthogonal view (axial, sagittal, and coronal) that independently process the input slices and make predictions. Each CNN returns predictions along the corresponding axis, indicating the presence of an organ at the specific slice. The prediction results are thresholded, and the largest 1D connected-components are extracted. These components are then merged to create a 3D bounding box per the structure of interest. Afterward, the obtained bounding boxes are propagated to finer levels of the multi-scale pyramid to operate with the data of higher resolution at the further analysis steps. In the following subsections, the method will be explained more thoroughly.

2.5.2. Input data preparation

The algorithm tries to predict classes $\boldsymbol{C}^{ax}$ of structures presented in every slice of the dataset $\boldsymbol{I}^{ax}$ along the specified axis ax, described as:

$$
\begin{aligned}
\boldsymbol{I}^{ax} &= \{I_0, I_1, \ldots, I_{N_{ax}-1}\} \\
\boldsymbol{C}^{ax} &= [\boldsymbol{C}_0, \boldsymbol{C}_1, \ldots, \boldsymbol{C}_{N_{ax}-1}] \\
\boldsymbol{C}_i &= \{1, 2, \ldots, C_{N_c}\} \\
i &\in \{0, 1, \ldots, N_{ax}-1\},
\end{aligned}
\tag{2.23}
$$

where each slice I_i has a corresponding vector of classes $\boldsymbol{C}_i$ represented by a set of integer values, which does not include the 0-th class, since it corresponds to the background. To provide slices and corresponding classes for the training procedure of the CNNs, the vectors of classes should be processed with the multi-hot encoding algorithm $f_{mhe}(\cdot)$:

$$
\begin{aligned}
\boldsymbol{C'}^{ax} &= f_{mhe}(\boldsymbol{C}^{ax}) \\
\boldsymbol{C'}^{ax} &= [\boldsymbol{C}'_0, \boldsymbol{C}'_1, \ldots, \boldsymbol{C}'_{N_{ax}-1}] \\
\boldsymbol{C}'_i &= [e_0, e_1, \ldots, e_{N_c-2}] \\
e_j &\in \{0, 1\} \\
i &\in \{0, 1, \ldots, N_{ax} - 1\} \\
j &\in \{0, 1, \ldots, N_c - 2\},
\end{aligned}
\tag{2.24}
$$

which produces a categorical vector $\boldsymbol{C}'_i$ of length $N_c - 1$, where presence or absence of a class C is denoted by 1 or 0 at the corresponding positions, which schematically depicted in Fig. 2.13. Thereby, complete absence of structures can be encoded as a zero vector of the same length, instead of reserving a specific vector element. This strategy allows unambiguously describe the content presented in the slices in the form suitable for the CNNs.

No structures
[0 0 0 0] [1 0 0 0] [1 1 0 0] [0 1 0 0] [0 1 0 0] [0 0 1 0] [0 0 0 1]
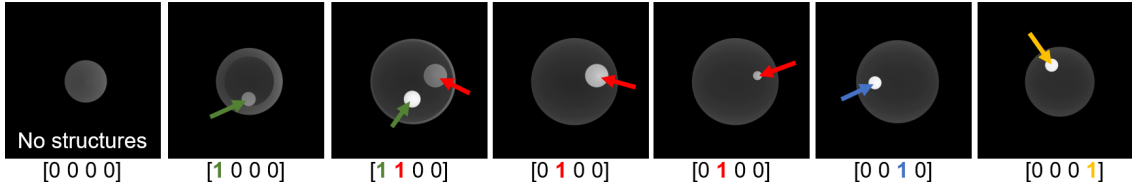
Figure 2.13.: The schematic representation of the multi-hot encoding process for several slices of a dataset, where different colors and activations correspond to presence of corresponding structures (green - Structure 1, red - Structure 2, blue - Structure 3, and orange - Structure 4).

The X-ray data is presented as a stack of slices describing a sample in a 3D space. Therefore, using 2D slices may not be efficient, because the third dimension providing depth is not taken into account. Hence, instead of 2D slices of size $W \times H$, the CNNs can be adapted to 3D slices of size $W \times H \times S$, where the extent of the third dimension is relatively small, a similar approach was described in Sec. 2.2.4.

Every slice along the specified orthogonal axis ax is accompanied with several neighbor slices N_{ns} taken before and after its position with a step of N_{ss} and form

a 3D spatial context $f_{ctx}(\cdot)$ for the central slice as:

$$\boldsymbol{I}^{ax}_{ctx} = f_{ctx}(\boldsymbol{I}^{ax}; N_{ns}, N_{ss})$$
$$\boldsymbol{I}^{ax}_{ctx} = [\boldsymbol{I}_0, \boldsymbol{I}_1, \ldots, \boldsymbol{I}_{N_{ax}-1}]$$
$$\boldsymbol{I}_i = [I_{i-j \cdot N_{ss}}, \ldots, I_i, \ldots, I_{i+j \cdot N_{ss}}] \qquad (2.25)$$
$$i \in \{0, 1, \ldots, N_{ax} - 1\}$$
$$j \in \{0, 1, \ldots, N_{ns} - 1\},$$

where $\boldsymbol{I}_i$ is a 3D slice at i-th position in the dataset along the specified axis, whose corresponding categorical vector describes the content of its central slice. The prediction of class is performed only for the central slice, whereas neighbor slices serve only for a depth context. In some cases, the context slices cannot be taken due to exceeding the minimal or maximal index along the specific axis, e.g. the slices at 0 and $N_{ax} - 1$ positions, thus, this issue can be resolved by replacing them with the central slice.

2.5.3. The network architecture

The architecture of employed multi-label CNNs is similar to the one already described in Sec. 2.2.4. This architecture is a generalization of the network proposed in [161], and it allows tuning a range of parameters to find the best architecture configuration for the specific application case.

The differences to already described architecture are that the network is aimed at prediction of N_c classes encoded with the multi-hot algorithm and the specific multi-label loss function [312] is used to obtain predictions for multiple labels simultaneously. This loss function is a special case of the cross-entropy loss function (Eq. (2.7)), where $N_c = 2$, because several classes in the slice are encoded by 0 and 1.

2.5.4. The prediction process

The multi-label CNN can be parameterized in a similar manner as it was done for the network at the validation stage (Eq. (2.8)). It accepts a dataset composed of 3D slices $\boldsymbol{I}^{ax}_{ctx}$ created from $\boldsymbol{I}^{ax}$ along the selected axis ax using a procedure described in Eq. (2.25). The network generates a prediction for each slice forming a prediction vector:

$$\boldsymbol{P}^{ax} = \text{CNN}^{ax}(\boldsymbol{I}^{ax}_{ctx}; S_{norm}, N_{bf}, N_{fc}, N_{dp}), \qquad (2.26)$$

which contains probabilities of presence of the specific structures in every slice (Fig. 2.14c). These prediction vectors should be post-processed to obtain the multi-hot encoded representations of classes, which in turn can be used to determine the ranges of indices describing the locations of target structures along

the respective axis. Combining the obtained ranges of indices allow to obtain 3D bounding boxes of the corresponding target structures (Fig. 2.14d).
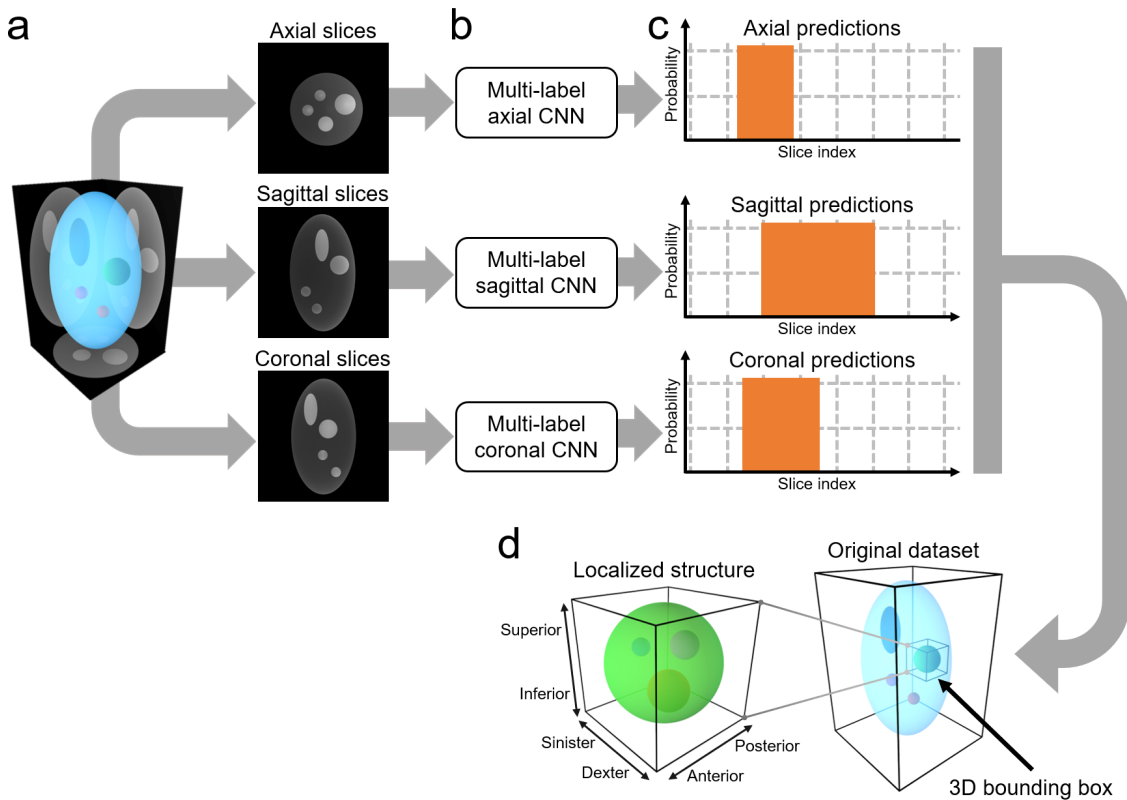
Figure 2.14.: The diagram depicts the localization process of target structures in a dataset (a) by deriving components of bounding boxes from axial probabilities vectors (c) using CNNs (b) trained on corresponding orthogonal views to obtain 3D bounding boxes (d) of respective structures.

2.5.5. The post-processing of predictions

The post-processing of the obtained prediction vectors $\boldsymbol{P}^{ax}$ is an essential step to deal with noisy results caused by abnormalities presented in the dataset. The prediction vectors obtained per orthogonal axis are smoothed with a Gaussian filter of σ_{pvs} to mitigate sharp peaks occurred at positions of slices carrying abnormal or ambiguous structures. The smoothed prediction vectors are then thresholded $f_{th}(\cdot)$ with some predefined value γ_{th} to obtain the multi-hot encoded representa-

tions of classes:

$$
\begin{aligned}
\boldsymbol{C}^{ax} &= f_{th}(\boldsymbol{P}^{ax}; \gamma_{th}) \\
\boldsymbol{C}^{ax} &= [\boldsymbol{C}_0, \boldsymbol{C}_1, \ldots, \boldsymbol{C}_{N_{ax}-1}] \\
\boldsymbol{C}_i &= [e_0, e_1, \ldots, e_{N_c-1}] \\
\boldsymbol{C}^{ax}[j] &= [\boldsymbol{C}_{0,j}, \boldsymbol{C}_{1,j}, \ldots, \boldsymbol{C}_{N_{ax}-1,j}] \\
e_j &\in \{0, 1\} \\
\boldsymbol{C}_{i,j} &\in \{0, 1\} \\
i &\in \{0, 1, \ldots, N_{ax} - 1\} \\
j &\in \{0, 1, \ldots, N_c - 1\}.
\end{aligned}
\tag{2.27}
$$

These binary vectors are subjected to the connected-component analysis $f_{cca}(\cdot)$ to extract the largest connected-component for j-th structure along the ax orthogonal axis:

$$
\begin{aligned}
\boldsymbol{R}_j^{ax} &= f_{cca}(\boldsymbol{C}^{ax}[j]; N_{ml}) \\
\boldsymbol{R}_j^{ax} &= [R_{j,0}, R_{j,1}, \ldots, R_{j,N_{caa}-1}] \\
R_{j,i} &= [r_{min}, r_{max}] \\
i &\in \{0, 1, \ldots, N_{caa} - 1\} \\
j &\in \{0, 1, \ldots, N_c - 1\} \\
R_j^{*ax} &= \max(\boldsymbol{R}_j^{ax}) \text{ by length,}
\end{aligned}
\tag{2.28}
$$

where N_{caa} is the number of extracted components $\boldsymbol{R}_j^{ax}$ of the j-th structure, N_{ml} is a minimum size of a connected-component considered for processing, $R_{j,i}$ is a vector formed by r_{min} and r_{max} indices describing a contiguous sequence of slices, and $\max(\cdot)$ is an operator to determine the longest range of indices R_j^{*ax}, which presumably includes the j-th target structure along the specified orthogonal axis.

Combining the location information from the most extended ranges of indices R^{*ax} for each axis with properly calibrated γ_{th} and N_{ml} parameters allows reconstructing a 3D bounding box B_j of the j-th structure of interest. If the most extended sequence was not detected for at least one orthogonal axis, then the 3D bounding box will not be constructed, and the corresponding structure cannot be localized.

2.5.6. Propagation through a multi-scale pyramid

Once the 3D bounding boxes are constructed for N_c structures, they should be propagated through a multi-scale pyramid $\boldsymbol{H}$ to make possible to localize structures hierarchically. The bounding boxes created at the l-th level of the pyramid should be adapted to every of N_L levels with $f_{bscale}(\cdot)$, which scales coordinates

of a bounding box by the specified scale factor:

$$
\begin{aligned}
B_j^{l'} &= f_{bscale}(B_j^l; \gamma_{bds}) \\
\gamma_{bds} &= \gamma_{ds}^{(l'-l)} \\
j &\in \{0, 1, \ldots, N_c - 1\} \\
l, l' &\in \{0, 1, \ldots, N_L - 1\},
\end{aligned}
\tag{2.29}
$$

where $B_j^{l'}$ is the bounding box of the j-th structure at the l'-th level of the multi-scale pyramid $\boldsymbol{H}$, given the bounding box B_j^l of the same structure at the l-th level, γ_{bds} is a scale factor calculated from the pyramid downscale factor γ_{ds}, the current and the next/previous pyramid levels l and l', correspondingly.

2.5.7. Accessing data at various pyramid levels

When the 3D bounding boxes for N_c structures were propagated through the multi-scale pyramid, the data at the higher pyramid level can be readily accessed as:

$$
\begin{aligned}
\boldsymbol{I}_j^{B,l+1} &= \boldsymbol{H}^{B_j^{l+1}} \\
j &\in \{0, 1, \ldots, N_c - 1\} \\
l &\in \{0, 1, \ldots, N_L\},
\end{aligned}
\tag{2.30}
$$

where $\boldsymbol{I}_j^{B,l+1}$ is the localized data of j-th structure extracted with the bounding box B_j^{l+1} from the multi-scale pyramid $\boldsymbol{H}$ at the $(l+1)$-th pyramid level. Thus, the data at a lower or higher resolution can be accessed by addressing the necessary level of the multi-scale pyramid.

2.6. Segmentation of localized structures

Among the segmentation techniques reviewed in Sec. 1.4.4, the deep learning-based methods demonstrated the state-of-the-art performance in the segmentation of multi-structural medical CT datasets. Therefore, these methods can be adapted for analysis of sequences of complex heterogeneous datasets of similar nature obtained during μCT experiments.

The 2D CNN-based methods usually utilize a slice-by-slice strategy to segment volumetric CT or MRI datasets [205, 313, 314, 206]. For instance, Roth *et al.* proposed spatial aggregation of holistically-nested networks for segmentation of pancreas CT data [313]. Havaei *et al.* proposed two-pathway shallow networks with cascaded architectures to segment glioblastomas in brain MRI datasets [205]. The novel architecture called U-Net was proposed by Ronneberger *et al.*, which was a 2D FCN architecture [200] and provided an efficient dense pixel-to-pixel segmentation [207].

Although these 2D CNN-based methods showed vast improvement in the segmentation accuracy over the classical machine learning methods based on hand-crafted features, they were still not optimal for 3D data analysis as they could not use spatial information inherent to volumetric data. To approach volumetric methods, Roth *et al.* proposed to train a network on centered orthogonal planes of a 3D patch to integrate richer spatial information; however, it was still limited to 2D kernels [208]. Soon after, the real 3D CNN-based methods emerged, they allow extracting much more powerful volumetric representation across all spatial dimensions [209, 210]. Kamnitsas *et al.* proposed to use a 3D network consisting of dual pathways and training strategy based on a dense inference technique of image segment to reduce computational complexity [210]. After segmentation, a 3D conditional random field model was applied to refine the result. This approach showed outstanding performance on segmentation of lesions in multi-channel MRI datasets.

Recently, several 3D volume-to-volume segmentation architectures were proposed, namely 3D U-Net [211], V-Net [212] and VoxResNet [213]. The 3D U-Net is an extension of 2D U-Net [207], which consisted of a contracting path to abstract features and an expansive path to produce a full-resolution segmentation. However, layers of the same resolution of these paths are linked through skip-connections. The V-Net divided the architecture into stages, integrated residual connections and proposed the objective function based on the Dice coefficient. This new objective function was used instead of the traditional cross-entropy to handle class imbalance issue occurred due to the significant difference between the number of annotated positive and negative samples. The VoxResNet borrowed the ideas of 2D deep residual learning [315] and constructed a very deep 3D network, whose multi-modality input and multi-level contextual information allowed to produce brain segmentation from MRI datasets.

The prediction learned with CNNs are usually refined with classical methods to improve the segmentation accuracy. In a range of works were proposed to combine CNNs with graphical models such as Markov random fields (MRFs) [216, 316] and conditional random fields (CRFs) [215, 317–321] to refine the network output. Moreover, a classical level-set method was used by Cha *et al.* [214] to refine a prediction result of a urinary bladder. Chen *et al.* [322] and Dou *et al.* [323] proposed to use a fully connected CRF for refinement because it is capable of capturing complicated shapes with holes or tiny structures.

2.6.1. Overview

The novel network architecture proposed by Ronneberger *et al.* [207] allowed for obtaining accurate segmentations without the need of additional refinement steps. The idea is to provide a common contracting network with successive layers, where up-sampling operations replace max-pooling layers. These layers increase the resolution of the output until it reaches the size of the input layer.

It allows for more accurate learning by incorporating local information, which is transferred by concatenating features from the contracting path with the corresponding features of the expansive path. In this way, the network can propagate local context information to higher-resolution layers. As the expansive path in some way is symmetric to the contracting path, the resulted network architecture resembles a U-shape. The network is trained in the end-to-end fashion, so it produces the probability map at the output layer of the same size as in the input layer. The training is performed for each orthogonal view extracted from the prepared 3D datasets as illustrated in Fig. 2.15.
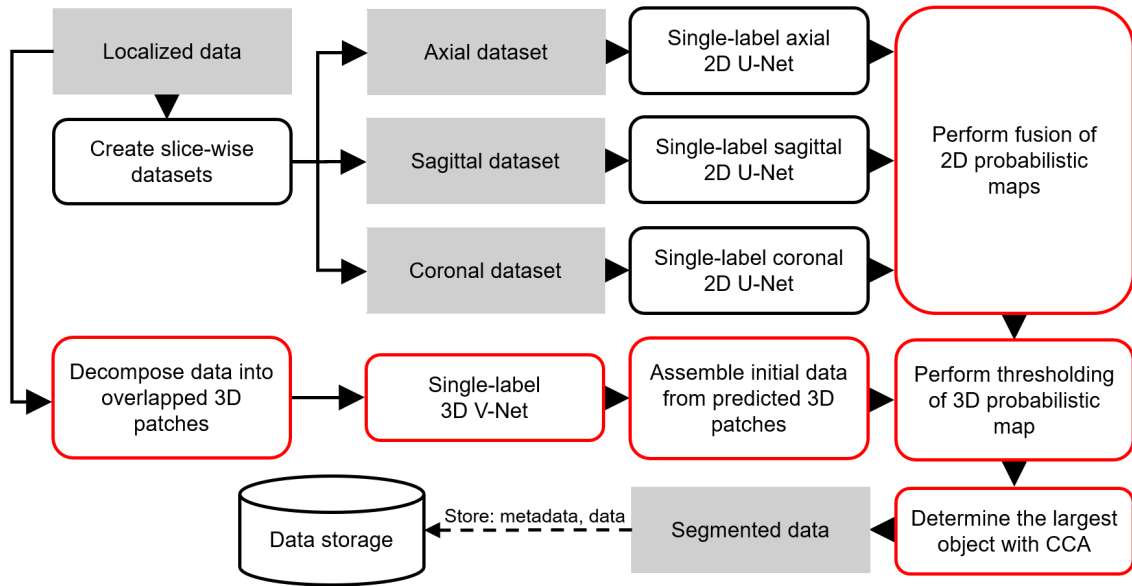

Figure 2.15.: The diagram of the segmentation stage which performs segmentation of a localized structure and post-processing. The red blocks represent new steps for two possible strategies. In the slice-wise strategy, axial 2D probabilistic maps are fused into a 3D probabilistic map, then it is thresholded, and the largest structure is extracted. The patch-based strategy uses 3D patches extracted from the dataset to train a network for further prediction of a 3D probabilistic map for each patch, assembling into the complete dataset, thresholding, and extracting the largest structure.

The resulted models produce 2D probability maps of the structure of interest along each of the axes. After that, these three stacks of probability maps are merged to generate more robust prediction by averaging probabilities for each voxel from all orthogonal views. In this section, the U-Net architecture [207] will be described and adapted for usage in the context of the high-throughput μCT experiments.

2.6.2. Input data preparation

The 3D bounding boxes B are estimated from the labels or obtained with the approach described in Sec. 2.5.1 are employed to extract the structures of interest for the dataset I:

$$I_j^B = I^{B_j}$$
$$j \in \{0, 1, \ldots, N_c - 1\}, \tag{2.31}$$

where I_j^B is the localized data of j-th structure with the bounding box B_j.

The input network size is fixed; therefore every input data must be prepared to fit the required size. However, due to the high memory consumption of the network, only a 2D U-Net is possible to apply per an orthogonal axis in a condition of restricted computational resources. The input data preparation can be divided into two strategies: the slice-wise and patch-based strategy.

The slice-wise strategy

In this strategy, all slices extracted along the corresponding axis are fed into the respective axial network with the specific input size. Since the input size of the network is fixed, the data should be scaled and uniformly zero-padded from all sides to fit the optimal size $S_{seg(j)}$ of the j-th structure as:

$$I_{sn(j)}^B = f_{snorm}(I_j^B; S_{seg(j)}), \tag{2.32}$$

where $I_{sn(j)}^{B(ax)}$ is the scaled localized data, whose sizes of slices along each orthogonal axis ax match the size of input data $S_{seg(j)}^{ax}$ of the corresponding network. Hence, the localized and spatially normalized data enclosing the j-th structure can be represented as:

$$I_{sn(j)}^{B(ax)} = [I_0, I_1, \ldots, I_{N_{ax}-1}]$$
$$i \in \{0, 1, \ldots, N_{ax} - 1\} \tag{2.33}$$
$$j \in \{0, 1, \ldots, N_c - 1\},$$

for each 2D U-Net trained on all slices along orthogonal axis ax, where I_i is a slice at the i-th position in the extracted data region $I_{sn(j)}^{B(ax)}$ along the specified axis.

The patch-based strategy

In the patch-based strategy, the network is trained on slices obtained along the axis ax of 3D patches of the fixed size S_{pth}, which are extracted from the localized and scaled data $I_{sl(j)}^B$ of the j-th structure as using $f_{pext}(\cdot)$ with the specified

overlap of γ_{ovlp}:

$$
\begin{aligned}
\boldsymbol{I}^B_{sl(j)} &= f_{scale}(\boldsymbol{I}^B_j; S^{ax}_{seg(j)}) \\
\boldsymbol{I}^P_j &= f_{pext}(\boldsymbol{I}^B_{sl(j)}; S_{pth}, \gamma_{ovlp}) \\
\boldsymbol{I}^P_j &= [\boldsymbol{I}^{P_0}, \boldsymbol{I}^{P_1}, \ldots, \boldsymbol{I}^{P_{N_{pths}-1}}] \\
\boldsymbol{I}^{P_k(ax)} &= [I_0, I_1, \ldots, I_{S^{ax}_{pth}-1}] \\
i &\in \{0, 1, \ldots, S^{ax}_{pth} - 1\} \\
j &\in \{0, 1, \ldots, N_c - 1\} \\
k &\in \{0, 1, \ldots, N_{pths} - 1\},
\end{aligned}
\tag{2.34}
$$

where $\boldsymbol{I}^{P_k}$ is the k-th 3D patch of size S_{pth}, and I_i is a slice of this patch taken along the selected axis ax. The procedure $f_{pext}(\cdot)$ extracts patches of size S_{pth} evenly across the complete dataset maintaining the overlap γ_{ovlp} to ensure smooth transitions between patches.

2.6.3. The training process

At the training phase, the network $\mathrm{UNet}(\cdot)$ is trained at the predefined subset of data $\boldsymbol{I}^{ax}_j$ and corresponding binary labels $\boldsymbol{L}^{ax}_j$ of the j-th structure:

$$
\begin{aligned}
\boldsymbol{I}^{ax}_j &= [I_0, I_1, \ldots, I_{N_{ax}-1}] \\
\boldsymbol{L}^{ax}_j &= [L_0, L_1, \ldots, L_{N_{ax}-1}] \\
i &\in \{0, 1, \ldots, N_{ax} - 1\},
\end{aligned}
\tag{2.35}
$$

where I_i and L_i are the i-th slices of data and labels obtained along the specified orthogonal axis ax. Then, these slices are fed into the corresponding network:

$$
\boldsymbol{W}^{ax}_j = \mathrm{UNet}^{ax}_j(\boldsymbol{I}^{ax}_j, \boldsymbol{L}^{ax}_j; S^{ax}_{seg(j)}, N_{bf}, S_{mrf}),
\tag{2.36}
$$

where $\boldsymbol{W}^{ax}_j$ is the set of trained network weights, $S^{ax}_{seg(j)}$ is the input data size along the corresponding axis for the j-th structure, S_{mrf} is the minimal size of the receptive field at the last convolution layer, and N_{bf} is the number of filters at the first convolution layer. The network performs an optimization procedure with a specified loss function Eq. (2.7) to learn the mapping from the data features to the output labels. However, depending on the strategy of input data preparation, the training procedure may slightly differ.

In a case of an insufficient number of training datasets, the data augmentation procedure may help to improve the robustness of the trained networks as described in 2.2.4.

The slice-wise strategy

The studied sample may vary in size due to peculiarities of a development process, which in the case of a living organism greatly influences the properties of internal structures. Due to these variations in properties, the location and size of the same structure in others datasets may be slightly different.

As the input size of the network is fixed, *i.e.*, all datasets are fed into the network should be of the same size. Therefore, the localized data containing the j-th structure should be spatially normalized to the size $S_{seg(j)}$, which is selected by averaging $f_{savg}(\cdot)$ among all sizes $\boldsymbol{S}_j^B$ of localized regions of the same j-th structure in other datasets:

$$S_{seg(j)} = f_{savg}(\boldsymbol{S}_j^B). \tag{2.37}$$

It is likely that the average size is not evenly divisible by two until the result of division reaches the specified size, as it is required by the network architecture. Therefore, each component of the size $S_{seg(j)}$ should be adjusted to obtain $S'_{seg(j)}$:

$$S'_{seg(j)} = f_{snear}(S_{seg(j)}; S_{mrf}), \tag{2.38}$$

which is evenly divisible by two until the size S_{mrf} is reached by looking for the nearest fitting value. In this way, it is guaranteed that the corresponding layers of the contracting and expansive paths will have the same sizes.

The patch-based strategy

In the case of the patch-based strategy, there is no need for additional adjustments of the size for the spatial normalization (*e.g.*, be divisible by two) or zero-padding to fit the specified size, because small patches are extracted from the data of much larger size. The localized data should be scaled while keeping the aspect ratio only to make structural content comparable in terms of scale with the similar structures from other localized data.

2.6.4. The architecture of U-Net

The U-Net architecture is illustrated in Fig. 2.16, where a contracting path is located on the left side, while an expansive path on the right. The first compresses the input data, whereas the second decompress the data until its original size is reached. All convolution operations are applied with appropriate zero-padding to avoid shrinking of the output and subjected to batch normalization to avoid internal covariate shift.
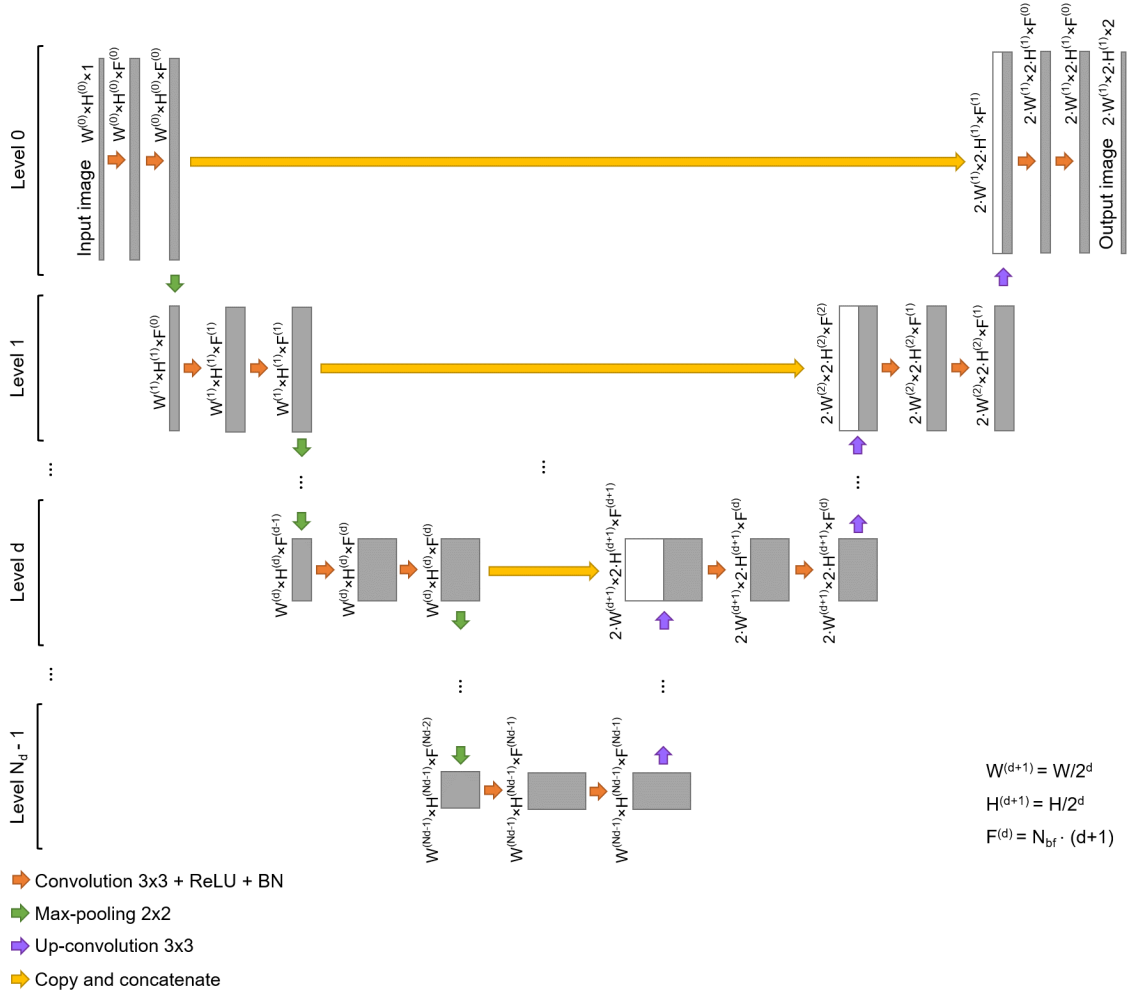
Figure 2.16.: The diagram of a generalized 2D U-Net architecture represented by contracting and expansive paths, where ReLU is an activation function and BN stands for a batch normalization technique.

The architecture of the compression path is arranged like in a regular CNN, and it is divided into several levels N_d operating at different resolutions. Each level comprises two convolutional layers with a kernel size of 3×3, equipped with a ReLU activation function followed by a max-pooling operation applied in non-overlapping windows of 2×2 to reduce the size of the current layer d by half $W^d/2 \times H^d/2 \times F$. Therefore, it is crucial that the size of the input layer $W \times H$ was selected in the way that the output size of each max-pooling operation has even values. The number of feature channels F is doubled at each level.

The decompression path implements a similar strategy but works in the opposite direction. It extracts features and expands the spatial support of the feature maps having lower resolution in order to collect and assemble the necessary information for the output layer. At each level of the decompression path, the up-sampling operation with the window size of 2×2 is employed in order to increase the size of the inputs, followed by a convolutional layer equipped with a ReLU

activation function using a halved number of 3×3 filters employed in the previous layer. Then, a concatenation with the corresponding feature maps from the compression path and two convolutions of the same configuration as in the previous layer is performed. At the last level, *i.e.*, the first level in the case of the decompression path, two feature maps are computed by a convolutional layer with a kernel size of 1×1 and a soft-max activation function to produce outputs of the same size $W \times H$ as the input layer and generate 2D probability maps of the foreground and background regions.

The forwarding of the features extracted at the levels of the compression path to the corresponding levels of the decompression path is illustrated in Fig. 2.16 by horizontal connections. This allows transferring the fine-grade details that would be otherwise lost during contraction of the feature maps that improve the prediction quality and the convergence time of model.

2.6.5. The prediction process

During the prediction phase, illustrated in Fig. 2.17, the slices from each orthogonal axis of the extracted data of the j-th structure are fed into the corresponding network equipped with the trained weights $\boldsymbol{W}_j^{ax}$. At the output, the network produces a collection of 2D probability maps $\boldsymbol{P}^{ax}$ for the specified axis ax:

$$\boldsymbol{P}_j^{ax} = \mathrm{UNet}_j^{ax}(\boldsymbol{I}_j^{ax}, \boldsymbol{W}_j^{ax}; S_{seg(j)}, N_{bf}, S_{mrf})$$
$$\boldsymbol{P}_j^{ax} = [P_0, P_1, \ldots, P_{N_{ax}-1}] \tag{2.39}$$
$$i \in \{0, 1, \ldots, N_{ax} - 1\},$$

where each slice P_i denotes the presence probability of the specific structure in every pixel.

In the case of the patch-based strategy, the resulted predictions for N_{pths} patches $\boldsymbol{P}_j^P$ extracted from the localized and scaled data of the j-th structure, must be combined with the help of $f_{pcmb}(\cdot)$ procedure:

$$\boldsymbol{P}_j^{ax} = f_{pcmb}(\boldsymbol{P}_j^P; S_{seg(j)}^{ax}, S_{pth}, \gamma_{ovlp})$$
$$\boldsymbol{P}_j^P = [\boldsymbol{P}^{P_0}, \boldsymbol{P}^{P_1}, \ldots, \boldsymbol{P}^{P_{N_{pths}-1}}]$$
$$j \in \{0, 1, \ldots, N_c - 1\} \tag{2.40}$$
$$k \in \{0, 1, \ldots, N_{pths} - 1\},$$

where the probabilities of the overlapped regions are averaged for every k-th patch $\boldsymbol{P}^{P_k}$ to reconstruct the 3D probability map $\boldsymbol{P}_j^{ax}$ composed of slices of size S_{seg}^{ax}.
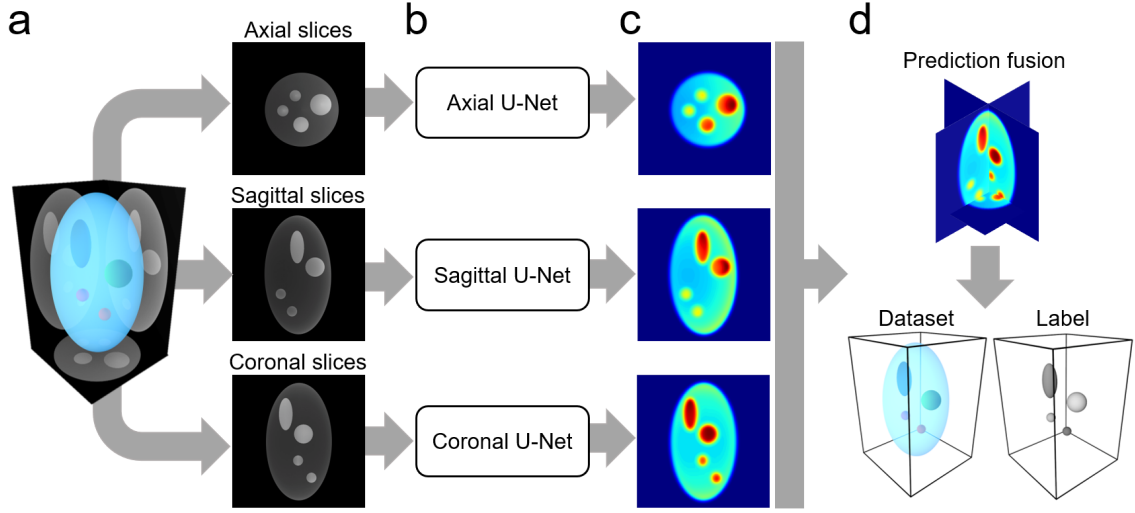
Figure 2.17.: The diagram illustrates the segmentation process of a target structure (a) using pre-trained U-Nets on corresponding orthogonal views (b) predicting 2D probability maps (c), which are then fused into a single 3D probability map (d) and thresholded to yield a binary segmentation.

2.6.6. The post-processing of predictions

The pseudo-3D probability maps $\boldsymbol{P}_j^{ax}$ obtained per orthogonal view ax for the j-th structure are represented as collections of 2D probability maps along the corresponding axes. Therefore, they should be fused to form the averaged 3D probability map providing a more robust prediction. Due to some parts of the target structure could be more perceivable from the specific orthogonal view, and less from others. These pseudo-3D probability maps are combined with $f_{fuse}(\cdot)$ by averaging probabilities at every voxel:

$$
\begin{aligned}
\boldsymbol{P}_{avg(j)} &= f_{fuse}(\boldsymbol{P}_j) \\
\boldsymbol{P}_j &= [\boldsymbol{P}_j^0, \boldsymbol{P}_j^1, \dots, \boldsymbol{P}_j^{N_{ax}-1}] \\
ax &\in \{0, 1, \dots, N_{ax} - 1\} \\
j &\in \{0, 1, \dots, N_c - 1\},
\end{aligned}
\tag{2.41}
$$

where $\boldsymbol{P}_{avg(j)}$ is the averaged 3D probability map across all axes for the j-th structure. The resulted probability map is then thresholded by the specified value of γ_{seg} to generate a binary segmentation of the j-th structure:

$$
\begin{aligned}
\boldsymbol{L}_j &= f_{th}(\boldsymbol{P}_{avg(j)}; \gamma_{seg}) \\
\boldsymbol{L}_j^{ax} &= [L_{0,j}, L_{1,j}, \dots, L_{N_{ax}-1,j}] \\
i &\in \{0, 1, \dots, N_{ax} - 1\} \\
j &\in \{0, 1, \dots, N_c - 1\},
\end{aligned}
\tag{2.42}
$$

where $\boldsymbol{L}_j$ is the 3D binary segmentation obtained by thresholding of the averaged 3D probability map, and $L_{i,j}$ is the i-th slice of the obtained segmentation for the j-th structure along the specified axis ax.

2.7. Morphological and statistical analysis

As it was said in Sec. 1.4.4, morphological analysis is highly important in diverse research fields such as medical, life and material science. The information about morphological properties helps to understand the microstructure of the studied sample and its physical properties. In its turn, it allows predicting the mechanical properties of the material to develop the new ones with the desired characteristics. Also, it helps to get insight into how structures change in a sample during the development or degradation processes.

2.7.1. Morphological analysis workflow

The general workflow is presented in Fig. 2.18, where the segmented input dataset L_j of the j-th structure is scaled back to the initial size to estimate the volume and the surface area properties correctly.
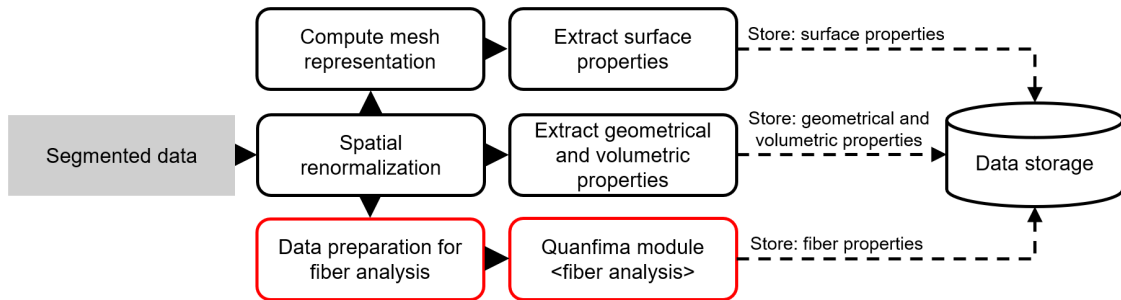

Figure 2.18.: The diagram of the morphological analysis stage which is used to extract various characteristics from segmentation of corresponding structures of interest.

The estimation of volumetric properties is performed by utilization of voxel information which is already presented by the label.

A polygonal representation also called as a mesh should be extracted from the segmentation L_j using the Marching Cube algorithm [324] to compute the surface properties. Then, the surface area may be calculated as a sum of polygon areas constituting the mesh. The compactness measures at this point may be estimated using the volume $\mathcal{V}_j$ and the surface area $\mathcal{A}_j$ values calculated for the segmented j-th structure.

The analysis of fibrous structures consists of the separate processing workflows aimed at the analysis of certain fiber properties. In the case of fiber orientation, it is composed of pre-processing, skeletonization, and successive morphological dilation of the segmented input data to provide more local support for the orientation estimation algorithms evaluated at each point of the skeleton. The detailed description and performance analysis of this workflow is provided in Sec. 3.1. Other types of structures can be also analyzed, but corresponding workflow and

underlying processing stages must be defined (*e.g.*, analysis of crack formation and crack growth in materials).

The estimated measures for the j-structure are saved into a persistent storage to be reused further.

2.7.2. Geometrical properties

The widely used method for estimation of the shape properties of particles is based on the American Society for Testing and Materials (ASTM) standard procedure [325] to measure the elongation and flatness ratios for various paving materials. As Fig. 2.19 illustrates, the elongation ratio may be represented by the length-to-width ratio, and the flatness ratio using the width-to-thickness ratio, where the length, width, and thickness are the object dimensions from the longest to the shortest.
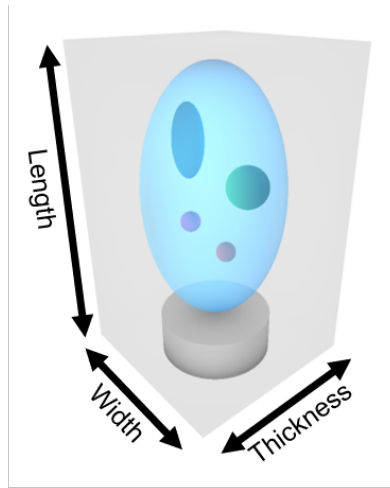

Figure 2.19.: The representation of elongation and flatness ratios of a sample or structure.

The 3D shape of the segmented j-th structure $\boldsymbol{L}_j$ is defined as the spatial distribution of voxels, whose properties can be analyzed with the central moments of order $(p + q + r)$:

$$\mu_{p,q,r} = \sum_{(x,y,z)\in\Omega} (x - x_c)^p (y - y_c)^q (z - z_c)^r \boldsymbol{L}_j(x, y, z) \qquad (2.43)$$

where $\boldsymbol{L}_j(x, y, z)$ is a value at a coordinate (x, y, z), and (x_c, y_c, z_c) is the centroid of the segmented structure. The central moments allows to determine geometrical and statistical properties of the structure. The zero-order moment describes the total volume of the structure, the second-order moments can be used to find the principal axes and orientation, whereas the moments of higher order describe rather the statistical properties of the distribution like skewness or kurtosis, than the geometrical ones.

2.7.3. Compactness and sphericity

The most basic measures for characterizing the geometrical shape of the segmented structure are the volume $\mathcal{V}$ and the surface area $\mathcal{A}$. The volume is calculated as the total number of non-zero voxels that comprise the studied structure, and the surface area may be estimated from a polygonal representation obtained from the segmented structure with the Marching Cube algorithm [324], which is the most popular method for surface extraction.

Along with the aspect ratio analysis, the shape of the structure can be measured regarding compactness, which is the dimensionless ratio of the surface area to the volume of the structure:

$$\mathcal{C} = \frac{\mathcal{V}}{\mathcal{A}}. \tag{2.44}$$

Moreover, the resemblance of the structure with a sphere can be estimated via the sphericity measure [326]:

$$\mathcal{S} = \frac{\mathcal{A}_s}{\mathcal{A}}, \tag{2.45}$$

where $\mathcal{A}_s = \sqrt[3]{36\pi\mathcal{V}^2}$ is the surface area of the sphere, containing the same volume as the structure. The sphericity is defined in the range $[0, 1]$, whereas the upper bound is reachable only if the studied structure is a true sphere.

2.7.4. Connected-component analysis and skeletonization

The connected-components analysis algorithm [327, 217] is the most widely used method for extracting groups of adjacent voxels, *i.e.*, connected-components. All voxels constituting a connected-component share similar intensity values and a particular type of connectivity. Once all groups are determined, each voxel is labeled with the specific intensity corresponding to the component it was assigned.

The skeletonization process [328, 329] reduces the binary regions of the studied structure to skeleton parts that largely preserves the extent and connectivity of the original structure while removing most voxels belonging to the original structure regions. One of the possible implementations may use morphological thinning that successively removes voxels from the boundary of segmentation and trying to preserve the ends of line segments until no more thinning is possible, which results in a skeleton.

2.7.5. Fiber analysis

The analysis of fibrous structures forms a separate research field applied in various areas of medical, material, and life sciences. It comprises plenty of methods developed for different aspects of analysis, such as estimation of density, orientation, diameter, length, connectivity, and other fiber properties in a 2D or 3D space. Due to the particular usefulness of the orientational analysis in the characterization of materials and tissues, the novel method for 3D orientation analysis will be presented in Sec. 3.1.

2.8. Implementation

This chapter introduces the programmatic architecture of the described analysis workflow presented as a set of modules combined into a framework. All general image processing operations were decoupled from the module-specific operations into a separate module. In this way, the modules perform only a single task and dependencies from other modules are minimized.

2.8.1. General architecture

The architecture of the segmentation framework is presented in Fig. 2.20 using Unified Modeling Language (UML) to explain the relationships between multiple modules. The framework consists of several internal modules, such as I/O Data, Validation, Processing, Alignment, Localization, Segmentation, and Analysis. Besides the internal modules, there are several external packages denoted here as external modules, namely, NumPy and sci-kit image. These modules are globally imported and used in all internal modules, along with locally imported ANTsPy, Keras, and quanfima modules providing the specific functionality for particular internal modules. The access to databases or storages, namely Large Data Storage, Fast Temporary Data Storage, and Meta-data Storage, was arranged through the I/O Data module to unify the interfaces because a saving process could be represented either by writing files on a disk or by adding new entries into a database. The framework was implemented using the Python language because it has a large collection of image processing and machine learning packages providing an easy interface to high-performance C/C++ implementations. The Python language allows for fast prototyping, widespread among the science community, and has a very calm learning curve to allow less experienced programmers to contribute into the code base. Moreover, it enables to use this framework in other Python-based environments by importing some of the modules.
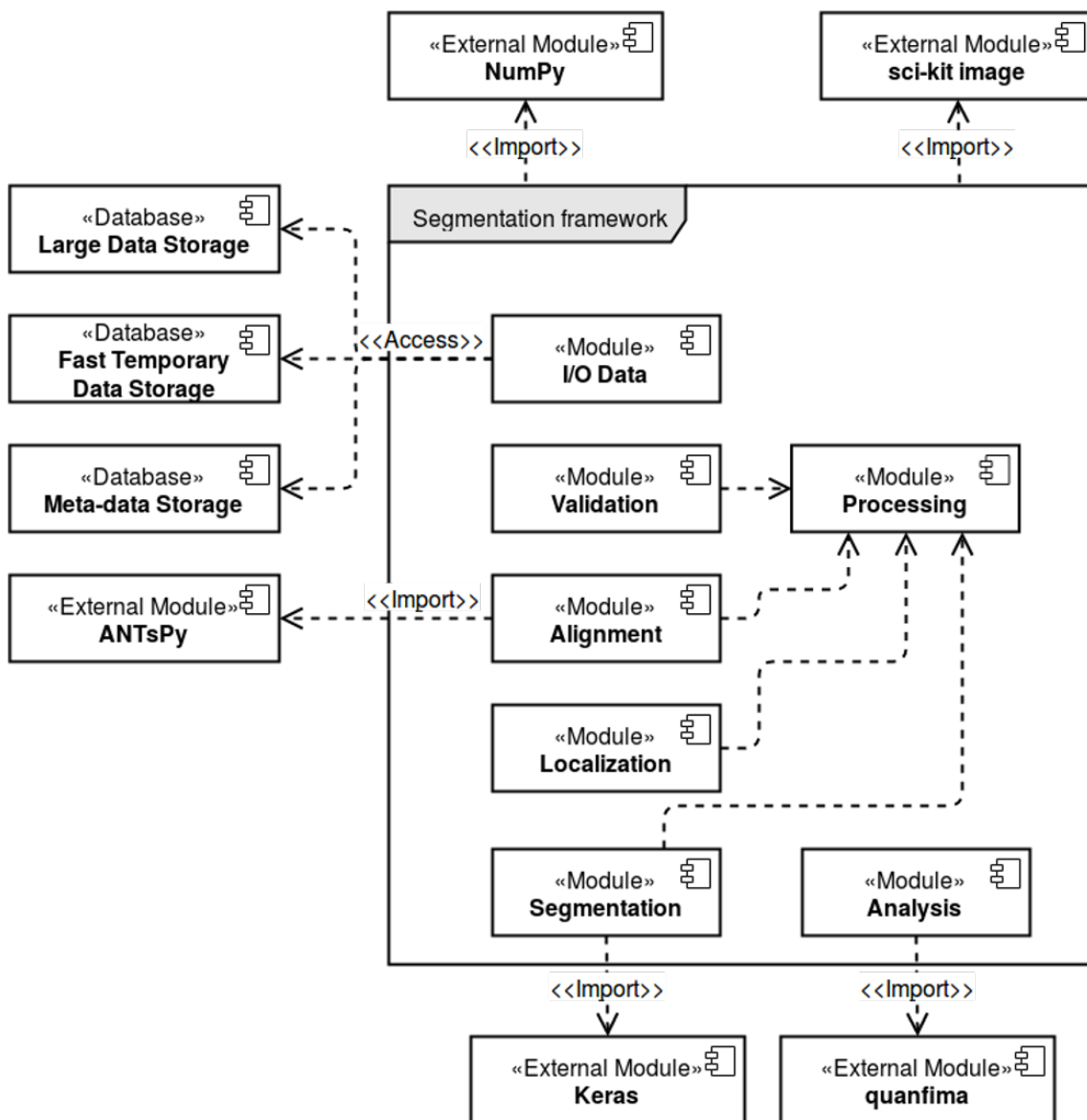
Figure 2.20.: The diagram of the segmentation framework composed of several modules, interacting with third-party modules and databases.

The emergence of IPython, and then Jupyter notebook made the creation of interactive web-based notebooks possible, where a user could write a code and immediately observe the execution result. These notebooks demonstrated that the code of some processing method could be attached to the paper, thereby closing the gap between the mathematical formulation and the practical in-code implementation. Therefore, the Python language provides a good opportunity for the integration with available data processing packages, large community and wide application in the research community. This framework was developed under the MIT license; therefore, it can be freely used in any other projects [330]. The detailed description of methods is presented in (Tab. B.15 to B.20).

In the following sections, only key methods from each module will be pre-

sented.

2.8.2. Validation module

The module implements the functionality for checking the suitability of the dataset for further usage in the analysis workflow. To distinguish between valid and invalid cases, the class `DataValidator` provides the method `train` to train the model at representative cases, which are passed into this method one after another. The suitability of the dataset for the further analysis is estimated by the method `validate`, which takes the dataset, and returns a Boolean value, `True` if the data is accepted and `False` otherwise.

2.8.3. Processing module

In this module, all image processing operations are gathered together to be accessible from one place for other modules. The data normalization routine with embedded data type conversion is available from the `normalize` function, which expects the dataset on the input. The optional arguments of this function are minimum and maximum intensities used in the normalization process, and the target data type employed in the data conversion.

The uniform zero-padding operation is provided by the `zeropad` function, which takes the dataset on the input and amount of zeros to add along every axis. The function returns the zero-padded dataset and the bounding box containing the original data.

To extract the largest connected-component from the binary data, the function `find_largest_object` is provided. It takes the binary data on the input and returns the extracted connected-component and its bounding box.

The function `rescale` scales the input data to the specified size and returns the original and rescaled data, and the scaling factor.

The averaging intensities at each voxel are accomplished with the `average` function, which takes a list of datasets of the same size. Then, these datasets are merged by averaging values at every voxel, and the averaged dataset is returned.

2.8.4. Alignment module

This module uses the ANTsPy package that provides Python bindings to Advanced Normalization Tools software [331], which is widely applied for the image registration tasks.

To align one dataset to another, the method `align` is provided by the class `Data Aligner`, whose arguments are the reference dataset and the target dataset, the transformation, the number of Gaussian pyramid levels, the minimized metric, and sampling strategy. The additional arguments are matched to the ones the antsRegistration application accepts [331]. The method returns the reference dataset, the aligned target dataset, and the matrix of the geometrical transformation.

Applying the existed transformation to the dataset may be done with the method `apply_transform`, by passing into the target dataset and the transformation matrix to apply. The method returns the original and the transformed dataset.

2.8.5. Localization module

The module implements the functionality to localize the structures of interest in the datasets using the deep learning approach. The localization model must be trained on the datasets with already annotated bounding boxes. To perform training, the class `DataLocalizator` provides the method `train` which takes the training and validation datasets, the number of layers and returns the trained model. The method `predict` allows for prediction of the bounding boxes by passing inside the target dataset and returns the prediction vectors, one per the structure.

2.8.6. Segmentation module

This module provides the functionality enabling to perform segmentation of the structures of interest using the CNN of the specific architecture. The method `train` of the class `DataSegmenter` takes the training and the validation datasets on the input, and several other parameters, such as the depth of network, the kind of loss function, the optimization method, and returns the trained model as the result. To apply the model to the previously unseen data, the method `predict` may be used. It takes the dataset as the input and returns a set of 2D probability maps, one per slice, depicting probability to find the target structure in every pixel.

2.8.7. Analysis module

To perform the morphological analysis of the segmented dataset, this module provides several routines. The volumetric and surface properties may be extracted with the functions `extract_vol_props` and `extract_surf_props`, which take the segmented dataset as the input argument, and return the lists of the extracted properties for each structure of interest. The fibrous properties are

extracted with the function `extract_fiber_props` which employs the quanfima package providing the functionality for the analysis of fibrous structures. The function returns the lists of the analyzed structures and the corresponding extracted properties.

2.9. Performance evaluation

The segmentation framework presented in this chapter should be evaluated on the series of synthetically generated datasets to systematically and quantitatively estimate optimal parameterization and accuracy for each stage. The secondary goal is to discover the influence of parametrization of some stages on others, the potential limitations of the entire system and its behavior in various noise conditions.

2.9.1. Dataset description

High-quality datasets with various internal structure are required to estimate the performance of the proposed segmentation framework quantitatively. Therefore, a synthetic phantom was designed [332] and used to generate 20 datasets (10 for a training and 10 for a validation/prediction population) for further comparative analysis, several representatives are depicted in Fig. 2.22. The phantom is presented by an ellipsoidal body composed of several ellipsoidal internal structures as illustrated in Fig. 2.21.
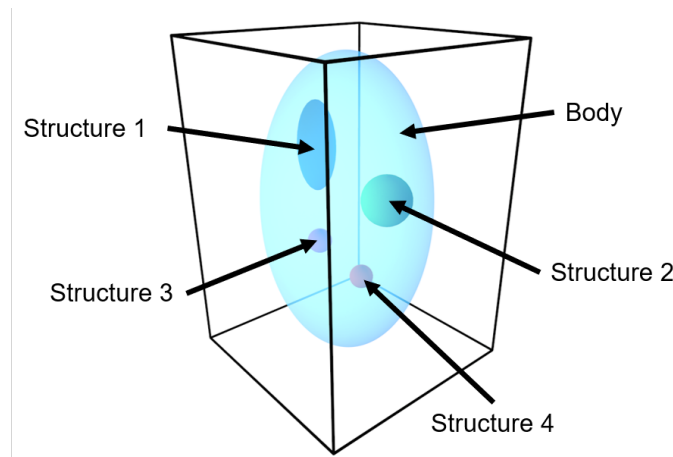

Figure 2.21.: The description of the phantom dataset composed of several nested ellipsoidal structures.

The properties of these structures such as location and size within the body are varying by the specified standard deviation to emulate variability in the real samples (*e.g.,* living organisms), the details are presented in Tab. 2.1. Every structure

was assigned a unique intensity value, which was manually selected in the way that all structures in the phantom have different intensities. The synthetic phantom allows evaluating the framework in non-trivial conditions by additionally contaminating datasets with the additive Gaussian noise to quantitatively estimate the behavior of each stage to quality degradation, which partially emulates insufficient exposure time during μCT experiments. Every generated dataset was

Structure	Type	Radius (voxels)			Location (voxels)			Value
		X	Y	Z	X	Y	Z	
Body	Ellipsoid	227±10	128±10	128±10	256	256	256	30
Structure 1	Ellipsoid	20±10	20±10	20±10	398±10	206±10	235±10	130
Structure 2	Ellipsoid	25±15	25±15	25±15	331±10	257±10	192±10	160
Structure 3	Ellipsoid	62±15	27±15	27±15	149±10	305±10	256±10	190
Structure 4	Ellipsoid	37±15	37±15	37±15	228±10	236±10	325±10	230

Table 2.1.: The configuration parameters of the phantom dataset.

smoothed with a Gaussian filter of σ_{smooth} of 0.5, and contaminated with the additive Gaussian noise σ_{agn} of 1.0 to rid of sharp transitions at the interfaces from the internal structures to the body and to introduce inhomogeneity, which is inherent to X-ray data. At the stages of evaluation of the robustness to the noise, the datasets allocated for evaluation were contaminated with the noise of level σ_{agn} in a range from 0.0 up to 40.0 with a step of 2.0. The datasets used in the training and prediction procedures are consistent through the whole evaluation process.
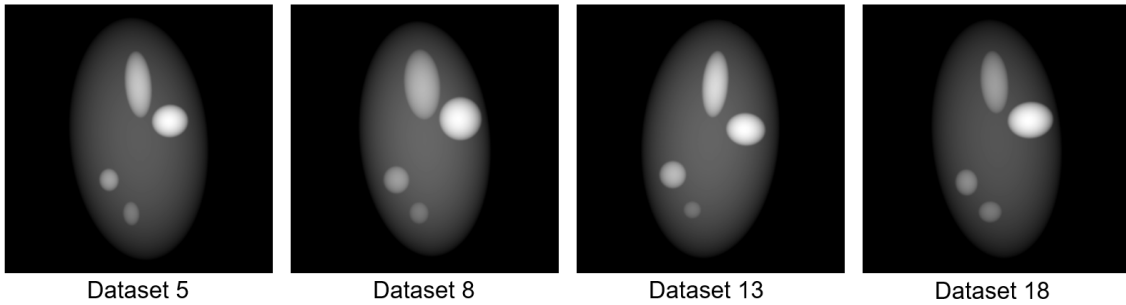

| Dataset 5 | Dataset 8 | Dataset 13 | Dataset 18 |

Figure 2.22.: The randomly selected representatives of the phantom datasets.

2.9.2. Benchmarking setup

The performance evaluation was done at a machine running 64-bit Ubuntu 16.04.5 LTS and equipped with 2 x Intel Xeon Silver 4114 processors, 4 x GeForce GTX 1080 Ti 12GB graphical adapters and 256 GB of random access memory (RAM). The datasets located at the Large Scale Data Facility (LSDF), and accessed via a local high-speed network of 320 MB/s.

2.9.3. Validation stage

The validation stage should be evaluated for the detection accuracy of anomalous samples to find the optimal network architecture and investigate how the noise influence on its accuracy.

Accuracy metric

The obtained predictions from the validation stage are first thresholded with the specified thresholding value to obtain binary predictions. The accuracy can be estimated using these thresholded values with the Dice score or so-called Dice ratio, which is widely used for estimation of segmentation accuracy:

$$\mathcal{DSC}(X,Y) = \frac{2\,|X \cap Y|}{|X| + |Y|}, \tag{2.46}$$

where X and Y are the obtained binary predictions and ground-truth labels.

Data preparation

Before to proceed with the validation process, the dataset should be quickly pre-processed by fast 2D median filtering of $W_{med} = 3$, which was selected smaller than the smallest structure presented in the dataset. The scaled size S_{val} of datasets was specified as 128×128×128, because it allows significantly reduce the data size, and subsequently the training and prediction time. The selected size should allow distinguishing between the general shape of the sample and the rough shapes of internal structures.

In the case of a lack of training data, the scaled data can be utilized to generate the new data by randomly translating the sample along all axes in the range from -64 to 64 voxels, and rotating around the center of the volume in the range from 0° to 45° for elevation and from -45° to 45° for azimuth. The transformed data is then assigned to the invalid class if the sample is out of the field of view or touches the volume boundaries, the valid class otherwise.

The transformed dataset is then converted into a sparse representation parameterized by the number of equally-spaced slices with a step of $N_{qss} = 10$, which was selected empirically. The complete parameterization is presented in Tab. B.1.

Training procedure

The optimal network architecture (Sec. 2.2.4) is selected with the help of the grid search procedure, which allows training networks on the grid of parameters created from Tab. B.2, and then select the most efficient architecture by the prediction accuracy on the validation datasets. The networks were parameterized with

various combinations of parameters as stated in Eq. (2.8): the number of base filters $N_{bf} = \{4, 8, 16\}$, the minimal size of the receptive field $N_{mrf} = \{4, 8, 16\}$, the dropout ratio of $N_{dp} = 0.5$ and the number of neurons $N_{fc} = 1000$ at the fully-connected layer. They were trained on the prepared training and validation datasets for 250 epochs using a batch size of 8 sparse representations. During each epoch, all datasets in a batch were passed through the network once. The initial weights of the network were set using the Glorot initialization. The Adam optimization [333] was used as a gradient-based optimization with learning rate $\alpha = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$. To avoid falling into a local minimum, the learning rate is reduced by a factor of 2 when a loss function was not improved in a row over 5 epochs. The training process stops when the value of the accuracy metric on the validation datasets stops improving within the last 10 epochs. In the end, the network with the highest accuracy on the validation dataset is selected as the final model.

Prediction procedure

The network produces a probability value that indicates the degree of validity of a given dataset, and then it is thresholded with the specified value to obtain a binary value, which corresponds to a valid or invalid class. The accuracy of the specific network architecture is calculated from a series of validation outputs from several datasets, which form a prediction vector. Next, the accuracy it is calculated with Eq. (2.46) from the produced prediction vector and the corresponding ground-truth vector.

Evaluation procedure

The evaluation procedure of the validation stage is aimed at the investigation of various network architectures to find the optimal one using the grid search method for a hyper-parameter optimization. The efficiency of the specific architecture completely depends on data; therefore, the only way is to brute-force some of the possible parameters with the assumption that the search happens in the vicinity of the optimal parameterization. Next, the optimally configured network should be estimated for the robustness of detection of datasets contaminated with various levels of the additive Gaussian noise.

The optimal network architecture

The validation networks were initialized with parameters specified in Tab. B.2 and trained without augmentation on the specially prepared datasets as described earlier.

The results of architecture benchmarking presented in Fig. 2.23 have shown that `CNN[4MRF 8BF]`, `CNN[8MRF 16BF]`, and `CNN[16MRF 4BF]` demonstrated

better accuracy than all other architectures, namely, 0.9786±0.009, 0.9798±0.0103, and 0.9808±0.0071, correspondingly. However, `CNN[16MRF 4BF]` provides the best detection accuracy, which is higher than `CNN[4MRF 8BF]` by 0.0022, and `CNN[8MRF 16BF]` by 0.001.
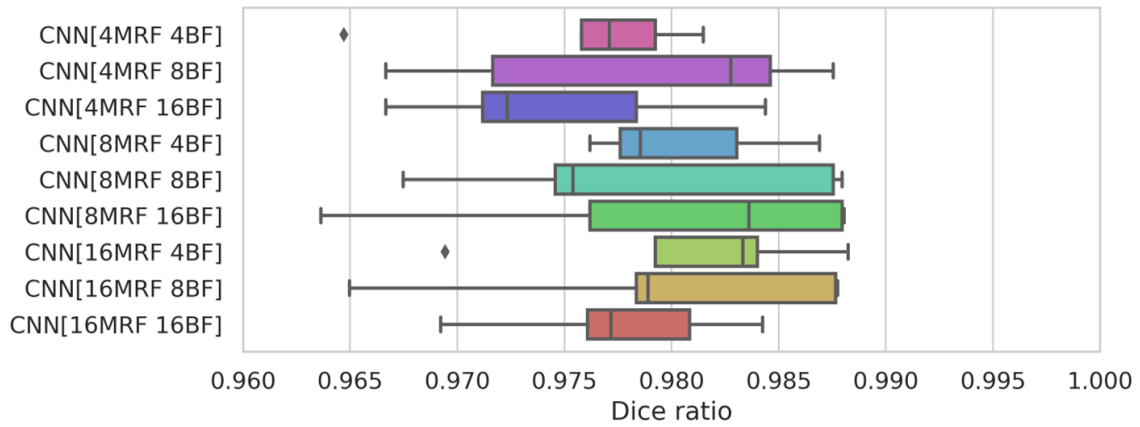

Figure 2.23.: The benchmarking results of the validation stage for determining of the optimal architecture.

Robustness to the noise

At this step, the optimally configured network should be evaluated for the robustness to the noise to determine acceptable SNR intervals of input datasets subjected to validation. These intervals allow rejecting the datasets, which have an excessive amount of noise and will not be adequately analyzed at the further stages.

The network was trained at the noiseless datasets from the training population. The five datasets from the validation population were contaminated with various σ_{agn} to estimate the robustness at each level of noise.

The Fig. 2.24 presents the results of robustness evaluation, demonstrating that the detection accuracy gradually decreases to zero starting from 6.25 of σ_{agn}. Thus, according to the obtained results, the values from 0.0 to 6.25 of σ_{agn} form a confidence interval, which corresponds to the interval from 13.13±0.14 to 12.38±0.13 dB of the SNR. This interval can be used in real experiments to determine datasets that will likely be analyzed correctly, given that their intensity distributions are similar to ones of the phantom datasets.
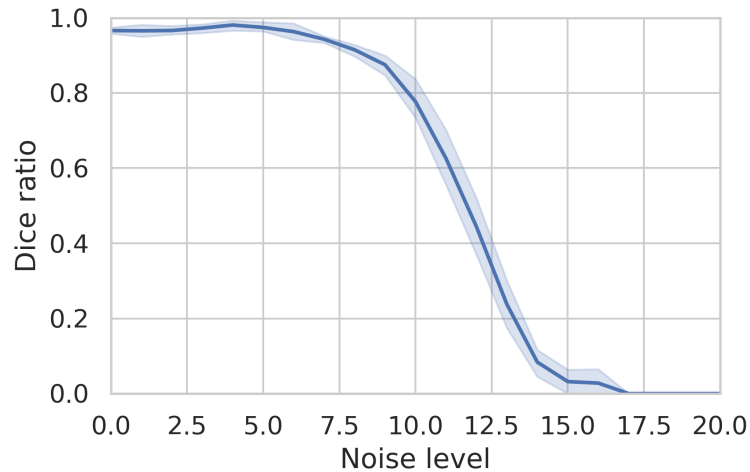
Figure 2.24.: The noise benchmarking results of the optimal network architecture for the validation stage.

Discussion

The quantitative analysis of the validation stage for the optimal network architecture has demonstrated the superiority of `CNN[16MRF 4BF]` over other considered network architectures. Furthermore, the selected architecture proved its resistance to the additive Gaussian noise up to 6.25 of σ_{agn}.

The selected optimal network architecture is parameterized by 4 filters at the first convolution layer, and the minimal size of the receptive field of 16 has provided higher prediction accuracy among other considered variants. The size of the receptive field of the layer located just before the fully-connected layer significantly influence the prediction accuracy. The small size of the receptive field promotes the elimination of most of the structural features by excessive downsampling provided by a max-pooling operation. In its turn, it leads to degradation of the discriminative power of features that in conjunction with a dropout process zeroing some of the feature values stimulates decreasing of the accuracy. On the other hand, the large size of receptive field provides much more feature values than the fully-connected layer can accommodate, which leads to ambiguity of the feature representation, and consequently worse accuracy of prediction. The number of features at the first convolution layer as well influence the representation of features, because the number of feature values at the layer before the fully-connected layer is the product of the number of filters and the size of the receptive field at the current layer. Thus, the large or small number of filters at the first convolution layer may potentially lead to either ambiguity or insufficiency of the representation of features. However, the only way to identify the best-suited configuration of the architecture is through a hyper-parameter optimization.

The recognizability of structures depends on the value of σ_{agn} or so-called the level of noise, whose source usually is low exposure time during the X-ray experiments. However, the visibility of structures expressed in the ability of the sample

to absorb the specific range of X-rays may also be controlled by the preparation protocol. The synthetic datasets were used to simulate various degrees of degradation with the additive Gaussian noise when at some noise level, the structures became no longer distinguishable for the network. It allows estimating the robustness of the network to potential outliers in the framework of the same experiment.

2.9.4. Data alignment stage

The efficiency of the registration procedure for data alignment was numerously presented in a range of works for the last decades. Therefore, here will be presented only the proof of principle that alignment can be performed for the synthetically generated datasets.

Accuracy metric

The accuracy of alignment may be estimated by calculating the MI metric Eq. (2.12) since it describes as one dataset explains another, and it maximizes when two datasets are co-aligned.

Data preparation

The synthetic datasets from the training population (10 datasets) are randomly rotated around the center of the volume in the range from 0° to 25° for elevation and from -25° to 25° for azimuth. In the case, when both orientation components are 0°, the phantom is co-aligned with the Z-axis. The rotated phantom is then randomly shifted along all axes from -20 to 20 voxels. All these transformations are required to simulate the case when the sample was improperly mounted on the rotation stage and started to move before the experiment started.

Method parameterization

The employed registration method was based on the rigid transformation model, which can describe an accidental movement of the sample before or during the experiment since it can only be either translated or rotated. The method was also equipped with the multi-resolution strategy to be able to process data of different size efficiently.

The registration model was initialized (Tab. B.3) with 4 zoom levels with corresponding scale factors of 8, 4, 2 and 1, where data at each level was smoothed using the Gaussian filter with σ_{smooth} of 3, 2, 1, and 0, correspondingly. At every zoom level, the optimization procedure using the gradient descent method

initialized with the mutual information cost function and the stopping criterion $\epsilon = 0.001$. This stopping criterion allows terminating the optimization process if the value of the cost function is not improved by more than ϵ from the previous iteration. This optimization process is run over 1000, 1000, 500, and 500 iterations per zoom level with scale factors 8, 4, 2, and 1, respectively. In the case, if the convergence occurs before reaching the limit of iterations, then the optimization stops and goes to the next level.

Evaluation procedure

The prepared datasets were registered to one of the reference datasets (Fig. 2.25) taken from the prediction population using the registration method described previously in (Sec. 2.3).
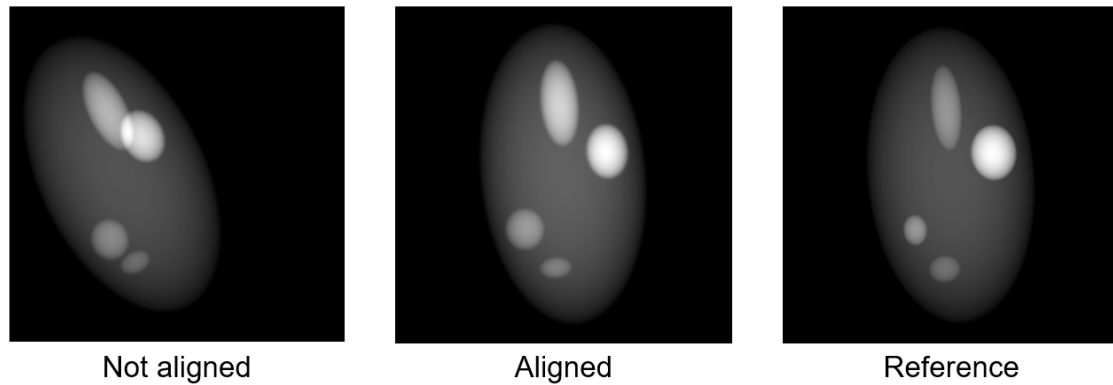

| Not aligned | Aligned | Reference |

Figure 2.25.: The alignment example of a randomly selected not aligned dataset to the reference dataset.

The results presented in Fig. 2.26 demonstrate that all aligned datasets have a small registration error Eq. (2.12) and visual assessment of a randomly picked registered dataset proves the correctness of the obtained results (Fig. 2.25).
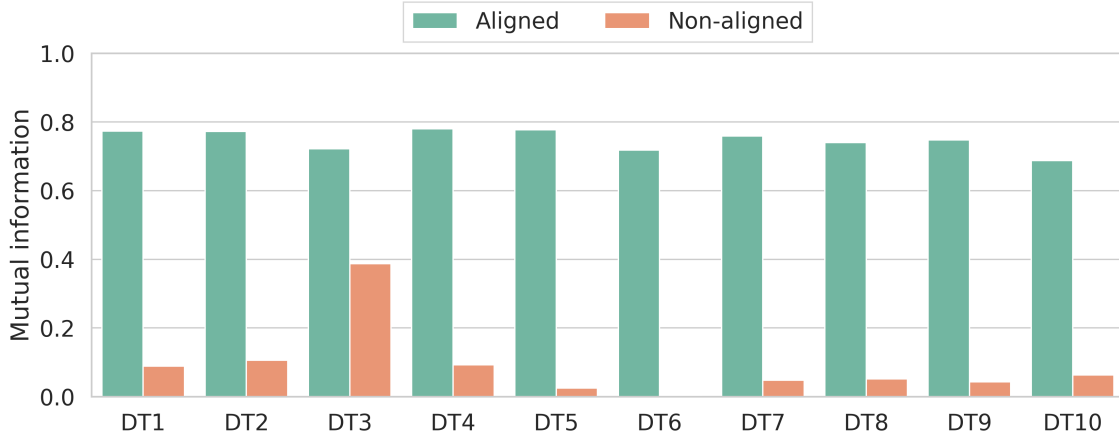
Figure 2.26.: The accuracy results of the alignment stage calculated in terms of the MI metric between the aligned and reference datasets (DT).

Discussion

The only difficulty of this stage is that the quality of alignment highly depends on the content of datasets, *i.e.*, in a case of completely different datasets, the process likely provide the incorrect result. Therefore, the accuracy estimation was performed for visually similar datasets, which are generated with the special phantom model, whose internal structures may vary in size and location within the predefined ranges. It is assumed that the datasets are similar to one another, and the structure of one may potentially describe the structure of another.

The obtained different metric values in the registration results can be explained by slightly different locations of the internal structures of the datasets. Therefore, it is impossible to obtain the single maximal metric value for all datasets, because the corresponding structures always differ in size and location within a population. It is only possible to maximize the metric value until it stops to change, that will point to that one dataset is aligned relative to another.

2.9.5. Pre-processing stage

The datasets should be pre-processed (Sec. 2.4) to suppress various types of noise (Eq. (2.16)), extract the object of interest from the volume (Eq. (2.17) to (2.20)) and create a multi-scale pyramid of the extracted object (Eq. (2.21) and (2.22)) to prepare it for the further analysis. The pre-processing stage was configured as in Tab. B.4 to proceed with the evaluation process. The NLM denoising step was initialized with the smoothing parameter $\gamma_{sm} = 7$, the patch size $W_{tp} = 3$, and the size of the search window $W_{sw} = 21$, which were determined by visual inspection of results obtained for various combinations of parameters. The sequence of morphological operations improving the global segmentation was parameterized by

the size of the structural element $S_{se} = 5$, which was selected based on the potentially smallest observable structure and the number of iterations $N_{mi} = 3$ was selected empirically. Then, the datasets were spatially normalized to the nearest size S_{norm}, which is divisible by two. The process of multi-scale pyramid creation was configured with the number of levels $N_L = 4$. This value was selected based on the minimal size of a dataset, so that, its internal structures still can be perceivable. The downscale factor of the multi-scale pyramid $\gamma_{ds} = 2.0$ is better to select in the way that it will be multiple of two, because all further stages will benefit from it, and finally the smoothing $\sigma_{pyr} = 1.0$ at each level is calculated relative to the smoothing parameter of the first level.

2.9.6. Localization stage

The localization should be evaluated on the accuracy of detection of 3D bounding boxes for the selected target structures. The goal is to investigate how the accuracy of detection degrades by increasing the amount of noise for the case of the optimal parameterization estimated for the noiseless datasets.

Accuracy metric

The localization stage predicts walls of a 3D bounding box enclosing the structure of interest. The prediction accuracy concerning the corresponding ground-truth 3D bounding box can be assessed by the wall distance measure $\mathcal{WD}(\cdot)$. This measure estimates the average distance between the corresponding walls of the predicted and the ground-truth 3D bounding box:

$$\mathcal{WD}(B_t, B_r) = \frac{1}{N_w} \sum_{i=0}^{N_w} \left| B_{t(i)} - B_{r(i)} \right|, \tag{2.47}$$

where B_t and B_r are the bounding boxes for the target and reference structure and N_w denotes the number of walls composing a bounding box.

Training procedure

Similar to the validation stage, the grid search method is used to determine the optimal network architecture for each orthogonal axis from a set of parameters Tab. B.5 by the wall-distance error Eq. (2.47). The networks were configured with the same parameters as in the case of the validation stage (Sec. 2.9.3). Every network was trained on the training and validation datasets prepared for the corresponding orthogonal axis as described in (Sec. 2.5.2) to provide the specific slice configuration as listed in Tab. B.5 with $f_{ctx}(\cdot)$, where the number of neighbor slices $N_{ns} = \{3, 5\}$ and the step between them is $N_{ss} = \{1, 2\}$. The training lasted for 250 epochs with a batch size of 8 contextual slices and used the same optimization technique as in Sec. 2.9.3.

Prediction procedure

Every network predicts the walls of the bounding box for each structure along the specified axis. The produced predictions are then converted into 3D bounding boxes as described in (Sec. 2.5.5). The bounding boxes are calculated for every structure in every dataset subjected for evaluation, and then they are used to estimate a wall-distance error Eq. (2.47) to the ground-truth bounding boxes. The errors obtained across structures of a dataset are averaged and then are averaged within a population of datasets as well. The final average wall-distance error is used as the accuracy measure for the specific network architecture.

Evaluation procedure

The performance analysis of the localization stage was conducted in several phases. At the first phase, the optimal parameterization should be found by a grid search method, while it completely depends on data, and in a case of different datasets, the parameters will differ. The obtained optimal parameterization is then used in conjunction with different augmentation strategies: no augmentation; the rigid strategy (R) is composed of translation and rotation; the affine strategy (A) includes the rigid and additionally performs scaling, and the piece-wise affine strategy (PA), which includes the affine with piece-wise deformations, whose parameters are described in Tab. B.6. The evaluation with augmentation is important because it allows understanding, how the optimally parameterized architecture will behave in conditions of a lack of data, and which augmentation strategy provides the best gain in accuracy. When the optimal parameterization with the best-suited augmentation strategy is found, then it can be evaluated on the sensitivity to the noise. The same datasets contaminated with different noise levels are passed through the network to discover at which level of noise it starts failing to determine bounding boxes properly.

The optimal network architecture

The localization networks were parameterized as listed in Tab. B.5. Then, they were trained on datasets generated according to the phantom model described in (Sec. 2.9.1) without involving any data augmentation strategy.

The results of the architecture evaluation presented in Fig. 2.27 demonstrate that `CNN[8MRF 16BF 3NS 2SS]`, `CNN[4MRF 8BF 5NS 2SS]`, and `CNN[4MRF 8BF 5NS 1SS]` provide the smallest error distances among other architectures, 0.0414±0.1035, 0.0418±0.1023, and 0.0425±0.1045 voxels, correspondingly. However, `CNN[8MRF 16BF 3NS 2SS]` showed a less distance error by 0.0004 and 0.0011 voxels relative `CNN[4MRF 8BF 5NS 2SS]` and `CNN[4MRF 8BF 5NS 1SS]`; thus it was selected as the best architecture.

The conducted evaluation has shown that the `CNN[8MRF 16BF 3NS 2SS]` trained at the generated population of datasets has provided the best accuracy among the network architecture and the slice configuration parameters. Hence, it provides the optimal parameterization of the network for this particular model of the phantom dataset.
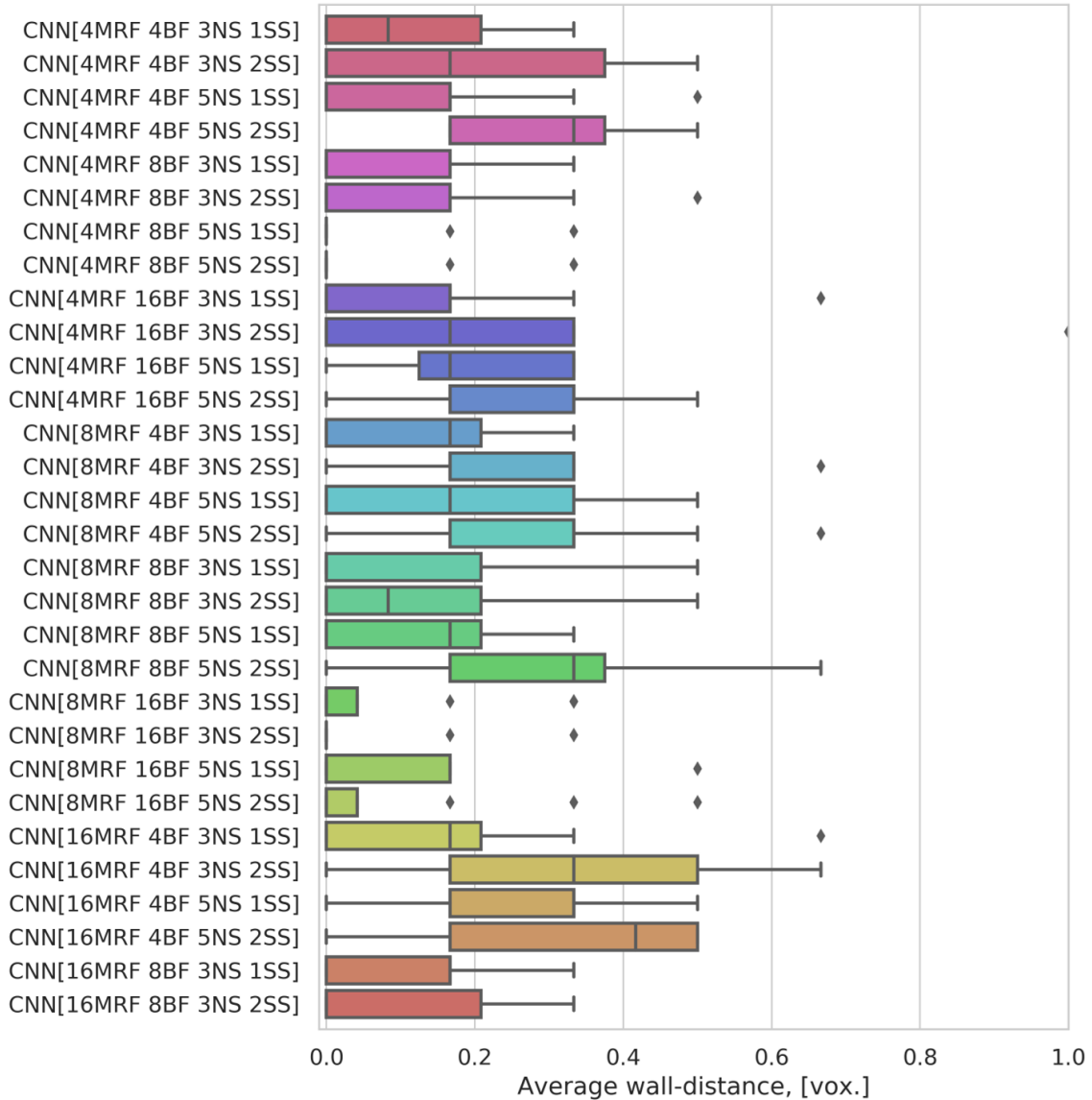

Figure 2.27.: The benchmarking results of the localization stage for determining of the optimal architecture.

The data augmentation strategy

As the deep learning methods require significant amount of training data, the generated datasets can be augmented to improve the accuracy of the networks. Therefore, the accuracy should be assessed for the networks utilizing the optimal

initialization parameters (*i.e.,* `CNN[8MRF 16BF 3NS 2SS]`) in conjunction with different data augmentation strategies participating in the training process.

The results are presented in Fig. 2.28, where the wall-distance error is shown for each structure separately for each network, and calculated as the average over the specific structure across all datasets participating in the prediction process. These results demonstrate that the network `CNN+R` has shown the best results among others for all structures, namely, it predicted the bounding boxes of Structure1 and Structure2 with a zero error, whereas other networks have insignificant errors, except `CNN+A` which has the error of 0.0500±0.0805 voxels for Structure2. While, in the case of Structure3 and Structure4, `CNN+R` has shown the errors of 0.0400±0.0644 and 0.0450±0.0724 voxels, 0.0500±0.0805 and 0.1000±0.1165 voxels for `CNN`, 0.0833±0.1178 and 0.0666±0.0860 voxels for `CNN+A`, and 0.0666±0.0860 and 0.0800±0.1032 voxels for `CNN+PA`, correspondingly. The mean errors for these structures relative to `CNN+R` are differ by 0.0100 and 0.0550 voxels for `CNN`, by 0.0433 and 0.0216 voxels for `CNN+A`, and by 0.0266 and 0.0350 voxels for `CNN+PA`, respectively.
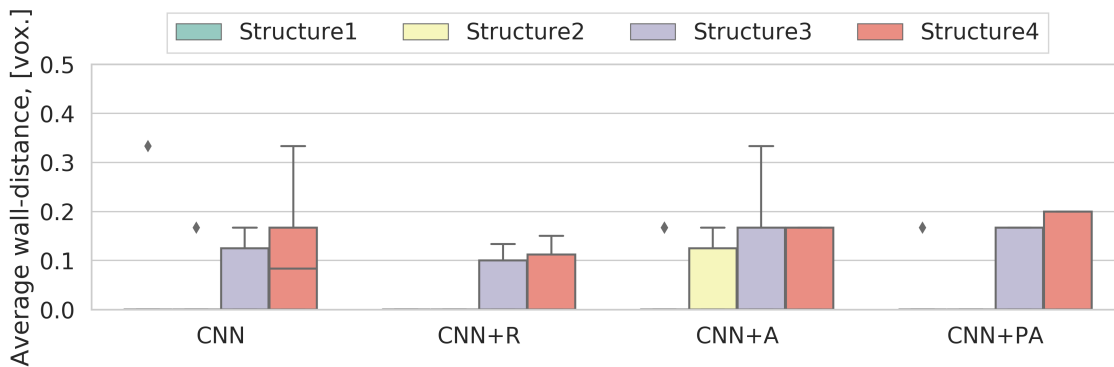
Figure 2.28.: The benchmarking results of the optimal architecture for various augmentation strategies of the localization stage.

Robustness to the noise level

When the optimal architecture and augmentation strategy of the network are found and successfully applied to the noiseless datasets, then its robustness to the datasets contaminated with the noise should be evaluated. It is necessary to determine the acceptable interval of the SNR and the noise level of the datasets, so that, internal structures be localizable.

The network configured with the optimal parameters and the augmentation procedure were trained using the same datasets as in the previous steps. Initially, the network was trained at the noiseless datasets. However, five datasets of the prediction population were sequentially contaminated with the specified noise levels to estimate robustness to every of them.

The obtained results presented in Fig. 2.29 demonstrate that the wall-distance error rapidly increases from 5 to 10 of σ_{agn}, while at higher noise levels, bounding boxes could not be created due to failing to localize structures at one or several orthogonal planes. It leads to the inability to extract the region containing the structure of interest and perform segmentation and further analysis of this structure. That allows determining the confidence interval from 0 to 5 of σ_{agn}, which corresponds to the interval of the SNR from 13.13±0.15 to 12.54±0.13 dB. Thus, the localization of internal structures in datasets having the SNR out of this range is likely to fail.
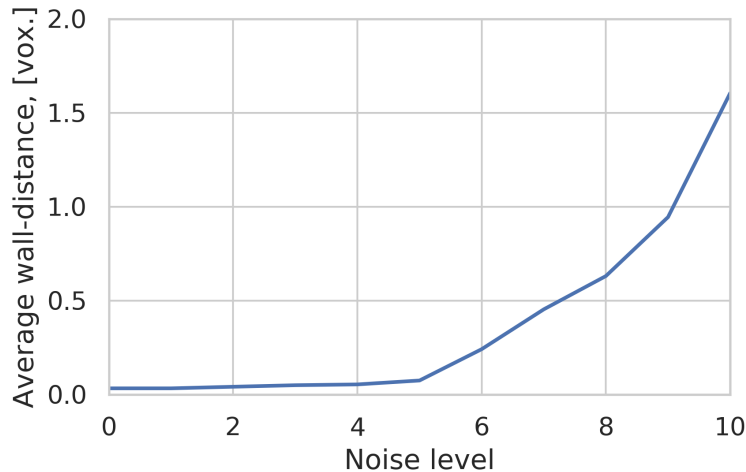

Figure 2.29.: The noise benchmarking results of the optimal network architecture with suitable augmentation strategy for the localization stage.

Discussion

The quantitative evaluation of the localization stage for the optimal network architecture and the data augmentation strategy has shown that the `CNN[8MRF 16BF 3NS 2SS]` network architecture with the rigid augmentation strategy, denoted by `CNN+R` provides the best performance among the considered configurations. Moreover, it demonstrated significant robustness to the additive Gaussian noise, starting failing only after 5 of σ_{agn}.

The obtained network architecture based on 16 filters at the first convolution layer and having a minimal size of the receptive field of 8 provides better accuracy than other considered architectures. It is because of the smallest size of the receptive field at the layer right before the fully-connected layer eliminates most of the structural features by excessive downsampling ensured by max-pooling operations. The consequences of such parameterization were previously described in (Sec. 2.9.3).

The network has shown the best accuracy results using a spatial context of 3 slices with a step of 2 slices between them. The accuracy mainly depends on the total number of slices forming a context, because it regulates how many neighbor

structures will be caught. If too many slices are selected for the spatial context, it might lead to ambiguity of prediction, especially at the borders of the target structure. This occurs since the network will be trained on slices which are out of the structure bounding box, and it will predict structure where it is absent. As for the step between slices, it influences in less degree, because it only regulates sparsity of the slices. However, if the step between slices is too large relative to the structure size, then prediction will fail because the integrity of the target structure will be violated and the network will be trained on the contextually disconnected slices.

The data augmentation was introduced into the training process to improve the accuracy by artificially increasing the amount of available data. The results have shown that the rigid augmentation strategy in conjunction with the `CNN[8MRF 16BF 3NS 2SS]` network architecture produces better accuracy than any other setup. It is connected to that the internal structures of the synthetically generated datasets change mostly rigidly among the generated population. All internal structures are represented with ellipsoids, which are parameterized by the radii and location, whose values are changing in the specified ranges, while the orientation is fixed. The body encapsulating all structures has fixed location; however, its size and orientation slightly change within the specified ranges. Therefore, all datasets are affected mostly by translation and rotation, which can be described by the rigid transformation model. As for changing the size of internal structures, it can be expressed via the scaling operation provided by the affine transformation model, however, its effect is too insufficient to use this model in the augmentation process. Concerning the piece-wise transformation model, it is too complicated for the case of the synthetically generated datasets since it elastically deforms the shape of structures. Taking into account that the synthetic dataset model is mainly driven by rigid transformation for its structures, usage of the piece-wise augmentation model will lead to training the network on the data, which cannot be produced by this dataset model.

The recognizability of structures on some extent depends on the level of noise, which source usually is low exposure time during the X-ray experiments. The synthetic datasets were used to simulate various degrees of degradation with the additive Gaussian noise to find the level when the structures become no longer be distinguishable for the network. It allows estimating the robustness of the network to the potential outliers occurring during the same experiment.

2.9.7. Segmentation stage

The segmentation accuracy of the selected target structures should be evaluated to investigate its dependency on the network architecture and the augmentation strategy. Also, the acceptable noise level of the input datasets should be determined to estimate the robustness of the optimally configured network.

Accuracy metric

In the case of the segmentation stage, it produces a 3D probability map, which is then thresholded to obtain the binary segmentation of the target structure. The automatically segmented structure must be compared to the ground-truth to estimate the algorithm accuracy. The most common measure for this purpose is Dice ratio Eq. (2.46), which is regularly employed in accuracy estimation of segmentation methods.

Training procedure

The optimal configuration of the U-Net architecture can be found similarly as in the cases of the validation and localization stage, using the grid search approach over the grid of parameters constructed from Tab. B.7. The networks were trained for each orthogonal axis separately on the prepared slices of the training and validation datasets along the corresponding axis. Every training process lasted for 250 epochs with the specified batch size of 8 slices using the same optimization method as for the validation (Sec. 2.9.3) and localization (Sec. 2.9.6) stages. The network architecture with the highest accuracy Eq. (2.46) on the validation dataset is selected per the corresponding axis.

Prediction procedure

The network predicts the presence of the target structure at every pixel of each slice along the specified axis. The predictions produced for each axis are fused to obtain a 3D probability map with the procedure $f_{fuse}(\cdot)$, which is then thresholded with $\gamma_{seg} = 0.5$ to yield the segmentation of the target structure. Afterward, the obtained segmentation should be cleaned up from potential artifacts and noise with $f_{cca}(\cdot)$, which appear in the result of thresholding. The cleaned up segmentation is then used to estimate the accuracy with Eq. (2.46) metric relatively to the ground-truth segmentation. The accuracy of the specific network architecture is calculated as the average of accuracies estimated for each dataset in a population, where the accuracy for a particular dataset is the averaged accuracy among all target structures.

Evaluation procedure

The evaluation of the segmentation stage consists of two phases, whose are similar to ones conducted in the evaluation of the localization stage. The first one is discovering the optimal network architecture with the help of the grid search approach because the architecture depends on the data. At the second stage, the best-suited data augmentation strategy among the rigid, affine, piece-wise approaches described in Tab. B.6 should be determined for the optimal network

architecture, since it potentially may improve the prediction accuracy. Finally, the network with the optimal architecture and training procedure should be assessed for the sensitivity to the noise. It is performed for the validation datasets, which are contaminated with various values of σ_{agn} to find out the noise level, at which the segmentation accuracy will start to degrade.

The optimal network architecture

The segmentation networks were configured as described in Tab. B.7 and trained without any augmentation strategy on the target structures obtained from the datasets generated with the phantom model described in (Sec. 2.9.1).

The results of the architecture evaluation are shown in Fig. 2.30, which illustrate that `CNN[8MRF 16BF]`, `CNN[4MRF 16BF]`, and `CNN[4MRF 8BF]` showed better accuracy among the rest architectures, namely, 0.9823±0.0047, 0.9821±0.0046, and 0.9818±0.0055, respectively. However, the network architecture `CNN[8MRF 16BF]` provides the best accuracy among them, which higher by 0.0002 and 0.0005 relative to `CNN[4MRF 16BF]` and `CNN[4MRF 8BF]`, correspondingly.
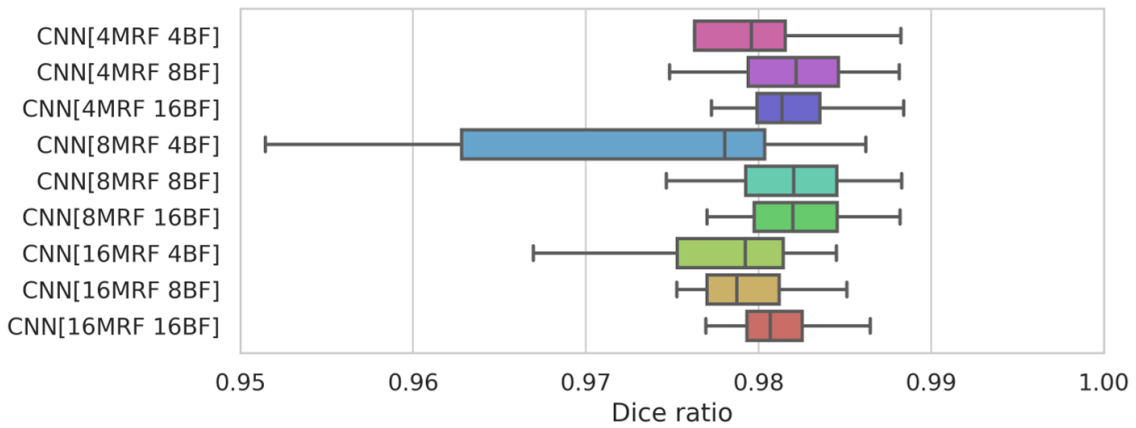
Figure 2.30.: The benchmarking results of the segmentation stage for determining of the optimal architecture.

The data augmentation strategy

As mentioned before in (Sec. 2.9.6), the training datasets can be augmented to improve the prediction accuracy of the networks. The networks initialized with the optimal parameters (*i.e.*, `CNN[8MRF 16BF]`) obtained in the previous section should be evaluated in conjunction with different augmentation strategies integrated into the training process.

The evaluation of the data augmentation strategies for two structures is presented in Fig. 2.31, where the averaged segmentation accuracy for each target structure

across all datasets involving into prediction is shown separately per augmentation strategy. According to the results, the network `CNN+R` provides the best overall segmentation accuracy of Structure2 and Structure3, namely, 0.9610 ± 0.0214 and 0.9814 ± 0.0069, whereas 0.8502 ± 0.117461 and 0.9823 ± 0.0056 in the case of `CNN+A`, 0.7378 ± 0.2133 and 0.9813 ± 0.0066 for `CNN`, and 0.8232 ± 0.1410 and 0.9816 ± 0.0058 for `CNN+PA`, respectively. The network `CNN+R` is superior in the average accuracy to other networks regarding to Structure2 by 0.2232, 0.1109, and 0.1379 for `CNN`, `CNN+A`, and `CNN+PA`, respectively. However, the network `CNN+R` is slightly inferior in the segmentation accuracy of Structure3 to `CNN+A` and `CNN+PA` by 0.0008 and 0.0002, correspondingly.

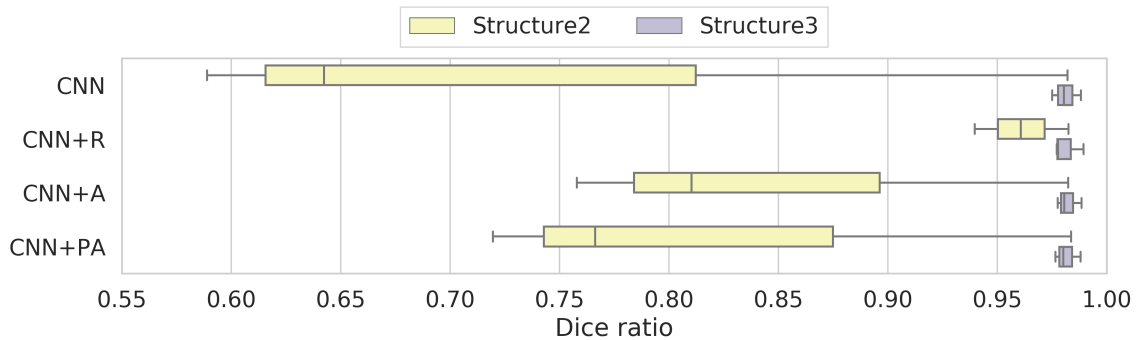**Figure 2.31.:** The benchmarking results of the optimal architecture for various augmentation strategies at the segmentation stage.

Noise robustness

The next step is to evaluate the network configured with the optimal parameters in conjunction with the proper augmentation strategy to the sensitivity to the noise. It allows determining the acceptable level of the noise, within which the accuracy almost does not change so that the target structures can be readily segmented.

The networks were trained on the noiseless datasets from the same population as in the previous steps. The five of the validation datasets were contaminated with the specified levels of noise to evaluate the robustness to each level.

The results of the robustness evaluation are presented in Fig. 2.32 and they demonstrate that the segmentation accuracy of Structure2 slowly starts to worsen right after 12 of σ_{agn}, and then continues to degrade until 35 of σ_{agn}. In the case of Structure3, its accuracy rapidly drops in the range from 12 to 18 of σ_{agn}. Hence, the confidence interval for both structures is from 0 to 12 of σ_{agn}, which corresponds to the SNR from 13.13 ± 0.15 to 11.11 ± 0.13 dB. Hence, the structures of similar datasets could be readily segmented if their SNR is within this range. The difference in the sensitivity to the noise of the structures can be explained by the size of the structure. It means that the smaller the structure, the higher the sensitivity, and hence, the harder to localize and segment it.
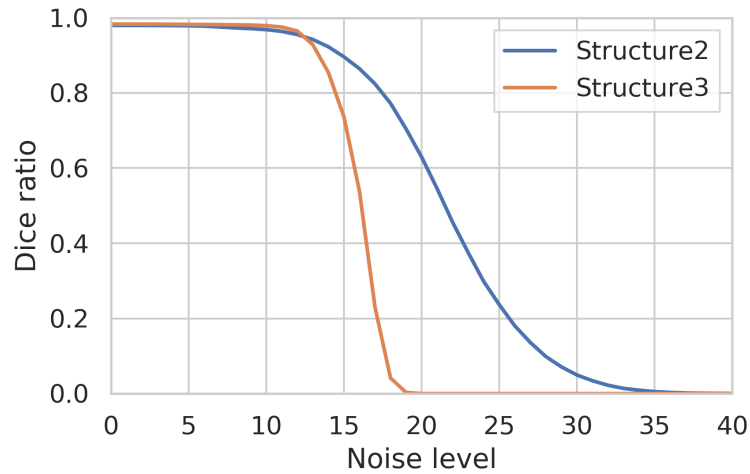
Figure 2.32.: The noise benchmarking results of the optimal network architecture with the suitable augmentation strategy for the segmentation stage.

Discussion

The performance evaluation of the segmentation stage for the optimal network parameters and the data augmentation strategy has demonstrated that the `CNN+R` network configuration produces the best results among others. Also, it proved the robustness to the additive Gaussian noise, maintaining the high accuracy up to 12 of σ_{agn}.

The selected network architecture provides better performance among the considered architectures, it is based on 16 filters constituting the first convolutional layer, and the network grows in depth until it reaches a minimal size of the receptive field of 8. The features are propagated through the contracting and expansion paths, where they are downscaling, and then upscaling to provide contextual features with the skip-connections. The tiny size of the last receptive field in the contracting path leads to feature vanishing due to excessive downsampling as it was described in (Sec. 2.9.6), while on the expansion path, these downscaled features provide worse distinguishability due to loss of feature information during downscaling. However, in the case of the large size of the receptive field, the features on the expansion path will not provide enough contextual features that will affect the final accuracy. The number of filters at the first convolutional level as well influence the performance, in such a way that a large number of filters may lead to overfitting, whereas a small number of filters to underfitting. These both cases lead to poor model generalization, and a result to low prediction accuracy. Therefore, the best-suited network configuration can be determined via a hyper-parameter optimization.

The rigid data augmentation strategy in conjunction with the optimal network architecture has provided the highest gain in the prediction accuracy among other considered strategies. Since the shape of the structures constituting the phantom

datasets is mainly defined by the rigid transformation model. Therefore, introducing more complex geometrical transformations into the training process will lead to teaching the network to segment structures of such shapes, which cannot be generated with the phantom model that will lead to poor prediction accuracy, as it was previously described in Sec. 2.9.6.

The analysis of the optimal network architecture with the best-suited augmentation strategy has demonstrated the robustness to the noise up to 12 of σ_{agn} allows resisting outliers which the SNR resides in the found noise intervals, more in detail is described in Sec. 2.9.6.

2.9.8. Summary

In this chapter, the segmentation framework for μCT datasets was presented. It is composed of various processing stages, such as data validation, sample alignment, pre-processing, localization, segmentation, and morphological analysis. The performance evaluation of each stage was done at the synthetic datasets with multiple different internal structures using the grid search method to find the optimal parameterization. It was shown that the framework could process a series of datasets of similar nature and provide persistent results with the optimal parameterization and the SNR level of the input datasets of at least 12.01±0.64 dB. The localization and segmentation stages allow determining bounding boxes and perform segmentation of small structures occupying less than 2% of the total object volume. In real-world applications, the segmentation stage can produce segmentations of complex structures requiring sophisticated approaches for analysis of internal morphology described in Chap. 3, since the analysis of simple geometrical properties could not be enough. Therefore, this framework can be used for real-world applications (Chap. 4) to analyze a large number of μCT datasets acquired under similar experimental conditions.

3. Orientation quantification method based on the ray-casting

3.1. Overview and related work

Quantification of fiber orientation is a common task in the fields of medical, material and life sciences. The orientation allows to determine the direction of fibers in tissues, predict properties of materials reinforced with fibers, validate and guide a fabrication process of scaffolds with controlled fiber orientation. Various methods were proposed to perform estimation of fiber orientation in a 3D space by analyzing a neighborhood of every voxel as described in Sec. 1.4.4.

The datasets can be acquired with different imaging methods (*e.g.,* light and electron microscopy, µCT, and others) that allow studying different aspects of a sample nature. However, among them, only the µCT imaging method has the largest penetration depth. It allows for a non-invasive investigation of an internal structure by collecting X-ray projections of the sample at different angles and subsequent reconstruction of cross-sections as described in Sec. 1.3.1. The size of the reconstructed µCT datasets usually reaches dozens of gigabytes due to the use of high-resolution detectors, which enable resolving more details while maintaining larger FOV.

The conventional workflow for orientation analysis of µCT datasets is composed of several stages: pre-processing, segmentation, a medial axis extraction, and orientation quantification. Different points of view on this workflow (Fig. 3.1) were presented in a range of works devoted to the analysis of fibrous structures in life and material sciences [225, 235, 334, 335].
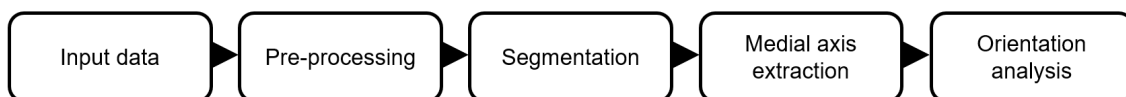
Figure 3.1.: The general workflow for fibrous orientation analysis.

The quality of acquired X-ray data varies and depends on parameters of an imaging method, an acquisition device, a sample preparation protocol and nature of a sample. Therefore, data should be pre-processed to correct noise with non-linear filters like median, bilateral or other edge preserving filters, in a case when the interested sample features are larger than a filter size, otherwise, they will be filtered out. On the other hand, data contaminated with beam hardening, streak,

and ring artifacts, described previously in Sec. 1.4.2, are very challenging and require specifically tailored approaches to be corrected. Moreover, these stages can be found in the previously described analysis workflow in Sec. 2.1.

The pre-processed data should be segmented to proceed with the further analysis. There is a vast of segmentation techniques which were mentioned in Sec. 1.4.4, however, their applicability depends on the nature of data. If fibers presented in the data are made of a homogeneous material and possess high contrast, then simple histogram-based methods might be applied. Otherwise, more sophisticated approaches based on machine learning are better suited.

After that, a medial axis or a skeleton should be extracted from isolated binary regions of the segmented data (Sec. 2.7.4). There are many algorithms for deriving the medial axis, some of them take into account crossing or touching regions [336–339].

When the medial axis of the segmented data is obtained, orientation estimation can be performed to determine the orientation of structures in the studied data. The most common and widespread method is based on the calculation of the second-order structure tensor which is used to quantify orientation from derived eigenvalues and eigenvectors. Despite it provides high-quality analysis, it is not rapid enough to process large μCT datasets in a reasonable amount of time due to a huge computation burden. Therefore, in this chapter, a new method for quantification of orientation will be presented.

3.2. Algorithms for orientation analysis

In this section, the proposed and the tensor-based methods are described in detail, along with a general workflow employed in orientation analysis as a prerequisite step, a fiber model in a 3D space, and the algorithm used to generate the synthetic dataset required for the further performance evaluation.

3.2.1. A fiber model

The fiber can effectively be modeled with a cylinder of length ρ, whose the center of mass is placed at the origin of the spherical coordinate system and oriented along the Z-axis, since orientation is estimated within a neighborhood of every point of its medial axis. Hence, an orientation of a bent fiber is an average orientation across all points constituting its medial axis. The orientation of the cylindrical fiber is defined by θ and φ angles determining tilts in different projection planes as presented in Fig. 3.2. The φ angle is elevation, which represents an inclination relatively the Z-axis, the θ angle is azimuth, specifying orientation in the XY-plane.
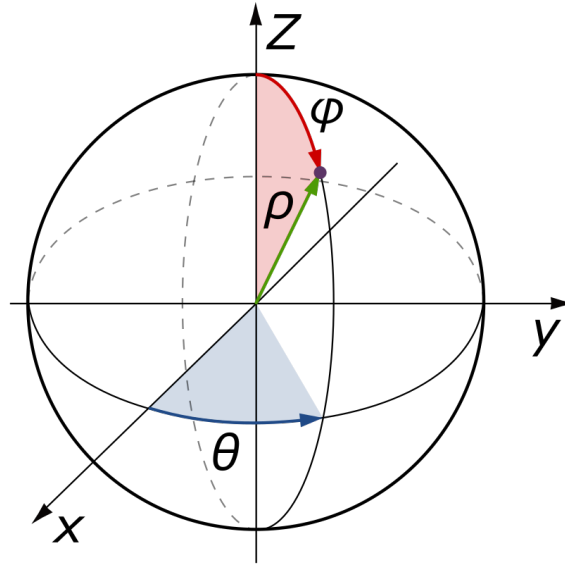
Figure 3.2.: The representation of a fiber in the spherical coordinate system.

It is assumed that angles vary from 0° to 90° for elevation and from -90° to 90° for azimuth, and both orientation components are 0° when the fiber is co-aligned with the Z-axis.

3.2.2. Tensor-based method

The second-order structure tensor $\hat{S}$ is estimated within a voxel neighborhood I' at every point of a medial axis of segmented fibers. The gradients I'_x, I'_y and I'_z along corresponding directions are calculated for every voxel of I' as:

$$\boldsymbol{K}_{yz} = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \boldsymbol{I}'_x = \boldsymbol{K}_{yz} * \boldsymbol{I}' \tag{3.1}$$

$$\boldsymbol{K}_{xz} = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \boldsymbol{I}'_y = \boldsymbol{K}_{xz} * \boldsymbol{I}' \tag{3.2}$$

$$\boldsymbol{K}_{xy} = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}, \boldsymbol{I}'_z = \boldsymbol{K}_{xy} * \boldsymbol{I}', \tag{3.3}$$

where $\boldsymbol{K}_{yz}$, $\boldsymbol{K}_{xz}$, and $\boldsymbol{K}_{xy}$ are the Sobel kernels extracting gradients along x, y and z axes, correspondingly.

Next, the smooth structure tensor matrix is constructed using the obtained gradi-

3. Orientation quantification method based on the ray-casting

ents convoluted with the specified smoothing kernel:

$$
\boldsymbol{S} = \boldsymbol{G}(0,0,\sigma_{tsr}) *
\begin{bmatrix}
I_x'^2 & I_x'I_y' & I_x'I_z' \\
I_x'I_y' & I_y'^2 & I_y'I_z' \\
I_x'I_z' & I_y'I_z' & I_z'^2
\end{bmatrix}
=
\begin{bmatrix}
\widetilde{I_x'^2} & \widetilde{I_x'I_y'} & \widetilde{I_x'I_z'} \\
\widetilde{I_x'I_y'} & \widetilde{I_y'^2} & \widetilde{I_y'I_z'} \\
\widetilde{I_x'I_z'} & \widetilde{I_y'I_z'} & \widetilde{I_z'^2}
\end{bmatrix},
\tag{3.4}
$$

where $\boldsymbol{G}(0,0,\sigma_{\mathrm{tsr}})$ is a Gaussian function with zero mean and σ_{tsr} standard deviation, defined as:

$$
\boldsymbol{G}(x,y;\sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x^2+y^2)}{2\sigma^2}}.
\tag{3.5}
$$

The smoothing is used to prevent cancellation of the gradient vectors with the same orientation but opposite direction. After that, the average tensor in the neighborhood $\boldsymbol{I}'$ is produced by averaging each element of matrix Eq. (3.4):

$$
\hat{\boldsymbol{S}} =
\begin{bmatrix}
\mu_{\mathrm{xx}} & \mu_{\mathrm{xy}} & \mu_{\mathrm{xz}} \\
\mu_{\mathrm{xy}} & \mu_{\mathrm{yy}} & \mu_{\mathrm{yz}} \\
\mu_{\mathrm{xz}} & \mu_{\mathrm{yz}} & \mu_{\mathrm{zz}}
\end{bmatrix},
\tag{3.6}
$$

where each element of the matrix Eq. (3.6) is the average gradient of the neighborhood $\boldsymbol{I}'$, calculated as:

$$
\mu_{ax} = \frac{1}{N \cdot N} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} x_{ij}
\tag{3.7}
$$

$$
ax \in \{xx, xy, xz, yy, yz, zz, xz\},
\tag{3.8}
$$

where N is the size of the element of the matrix $\hat{\boldsymbol{S}}$. Since, the resulted matrix Eq. (3.6) is positive semi-definite, its eigenvalues and eigenvectors can be obtained by solving $\left| \hat{\boldsymbol{S}} - \lambda \boldsymbol{I}_m \right| = 0$ as:

$$
\hat{\boldsymbol{S}} =
\begin{bmatrix}
\boldsymbol{v}_1^\top \\
\boldsymbol{v}_2^\top \\
\boldsymbol{v}_3^\top
\end{bmatrix}
\begin{bmatrix}
\lambda_1 & 0 & 0 \\
0 & \lambda_2 & 0 \\
0 & 0 & \lambda_3
\end{bmatrix}
\begin{bmatrix}
\boldsymbol{v}_1 & \boldsymbol{v}_2 & \boldsymbol{v}_3
\end{bmatrix}
\tag{3.9}
$$

$$
\lambda_1 >= \lambda_2 >= \lambda_3,
\tag{3.10}
$$

where $\boldsymbol{v}_1$, $\boldsymbol{v}_2$ and $\boldsymbol{v}_3$ are unit eigenvectors, and λ_1, λ_2 and λ_3 are corresponding eigenvalues. The $\boldsymbol{v}_1$ is orthogonal to the best fitting plane, $\boldsymbol{v}_2$ is orthogonal to the best fitting line within this plane and $\boldsymbol{v}_3$ is orthogonal to both and aligned with the direction of the least intensity change within $\boldsymbol{I}'$, whereas the eigenvalues specify the magnitudes of the corresponding eigenvectors. The tensor is based on intensity gradients estimated in all orthogonal planes; thus, the eigenvectors point to intensity changes in space. Hence, the primary orientation of the structures within the region $\boldsymbol{I}'$ is determined by the eigenvector $\boldsymbol{v}^* = (v_1^*, v_2^*, v_3^*)$ with the smallest eigenvalue λ^*, which is used to calculate the pair of angles as:

$$
\theta =
\begin{cases}
\tan^{-1} \frac{v_2^{*2}}{v_3^{*2}}, & v_3^{*2} > 0 \\
\frac{\pi}{2}, & \text{otherwise}
\end{cases},
\qquad
\varphi = \sin^{-1} \sqrt{v_2^{*2} + v_3^{*2}},
\tag{3.11}
$$

and points along the structures, where intensity variation is minimal.

3.2.3. Ray-casting method

The proposed method is based on the assumption that fiber orientation at each point (x, y, z) of the volume $\boldsymbol{I}$ can be determined by emitting rays from the corresponding point in all possible directions within its neighborhood $\boldsymbol{I}'$ (Fig. 3.3) and computing the related ray-sum (Eq. (3.12)). A ray-sum is calculated by summing intensities along a ray defined by a line L passing through a sub-volume $\boldsymbol{I}'$. The fiber direction (θ, φ) at each point is determined by the ray having a maximal ray sum:

$$f_{\text{ray}}(x, y, x; \theta, \varphi) = \int_L \boldsymbol{I}'(l \sin\theta \cos\varphi + x, l \sin\theta \sin\varphi + y, l \cos\theta + z) dl, \quad (3.12)$$

where $\boldsymbol{I}'$ is a sub-volume representing the neighborhood of the point (x, y, z) containing original values of volume $\boldsymbol{I}$, L is a line in the spherical coordinate system centered at the point (x, y, z), and $f^*_{\text{ray}}(x, y, z)$ is the primary orientation:

$$f^*_{\text{ray}}(x, y, z) = \underset{\theta, \varphi}{\operatorname{argmin}}\, f_{\text{ray}}(x, y, x; \theta, \varphi). \quad (3.13)$$

The propagation distance of the rays is limited by a sphere with a radius of $L/2$, centered at the point (x, y, z). The trade-off between the accuracy and the computation time can be achieved by adjusting the radius of the restricting sphere, the limits and the step of the angular scanning ranges.
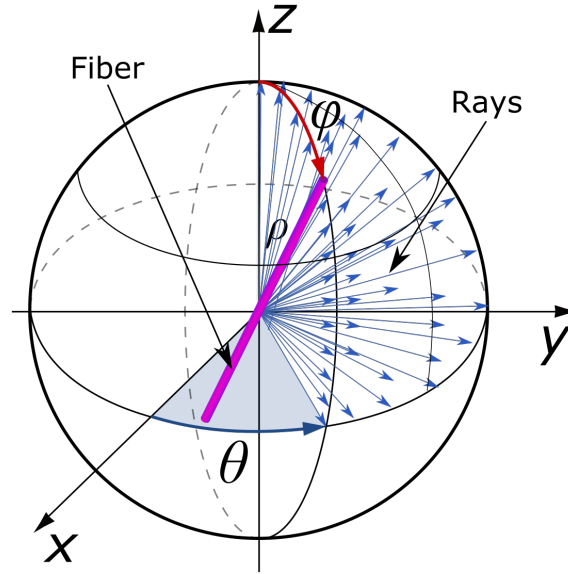

Figure 3.3.: The representation of a fiber in the spherical coordinate system with the rays casted from the origin.

3.2.4. Generating of a synthetic dataset

Synthetic datasets provide users with the ground-truth values (*e.g.,* orientation) which makes verification of imaging techniques or analysis algorithms possible.

3. Orientation quantification method based on the ray-casting

Here, the algorithm for the generation of non-curved fiber-like structures in a 3D volume is presented.

A fiber can be parameterized with a diameter value, a pair of tilt angles, azimuth and elevation, denoting orientation and a length value. The diameter and orientation values are sampled from given ranges, whereas the length of the fibers is specified as a percentage with respect to the minimum side length of the volume. Moreover, it is assumed that individual fibers do not cross each other and are separated by a specified gap value.

The fibers are added to the volume until a stop rule is satisfied. The stopping rule can be a maximum number of fibers, the ratio between the volume of all fibers and the total volume, or the number of failed additions due to intersections of fibers. Each fiber is initiated as a set of points representing a circle of the specified diameter. Then these points are rotated by the sampled azimuth and elevation angles (Fig. 3.4a) and propagated along the unit vector perpendicular to the profile (Fig. 3.4b). Then, the obtained coordinates are transformed to array indices and used for checking on the placing conditions and forming a fiber in a three-dimensional array (Fig. 3.4c).
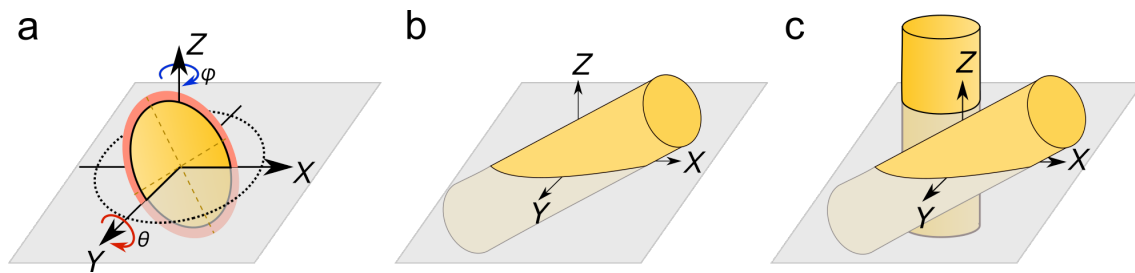

Figure 3.4.: The synthetic fiber generation process: a) the cross-section rotation in a plane of the 3D volume, the orange region represents a gap for a given fiber; b) the cross-section propagation along the specified direction to form a fiber; c) adding the fiber into the 3D volume.

To more closely simulate real fibers, the parameters of the fibers should be sampled from value ranges defined heuristically or analytically. If randomly oriented and tightly packed thin fibers in a small 3D volume are desired, the simulation process requires a high number of fibers, narrow distributions of the gap value and the diameter and a broad distribution of orientation values. The simulated volume consists of two materials, where zero values represent the background and non-zero values are synthetic fibers. The produced dataset can be contaminated with the additive Gaussian noise to emulate the noise produced by a digital detector. The noise value is sampled from the noise distribution and applied to every point of the 3D volume independently. Since physical dimensions are measured in pixels, thus fiber parameters can be controlled at the finest level and be versatile in terms of resolution. The maximum size of a simulated volume entirely depends on the amount of RAM of the specific computer. The example of the generated datasets in aligned, moderately aligned and disordered configurations are shown in Fig. 3.5(a-c).
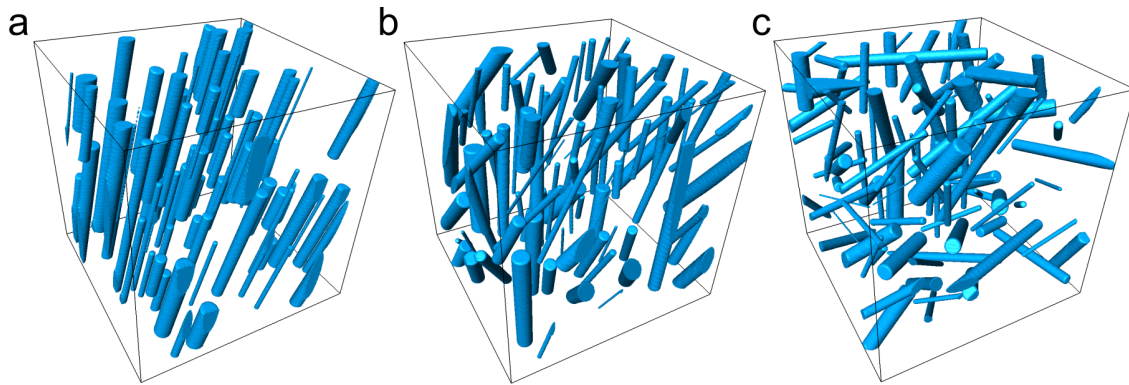
Figure 3.5.: The generated datasets with different fiber configurations: a) aligned; b) moderately aligned; c) disordered.

These datasets are composed of 100 non-intersected fibers, whose orientation of the aligned case is fixed to 27° and 15° for azimuth and elevation angle, respectively. The moderately aligned dataset has fiber orientations which ranges from -45° to 45°, and from 0° to 45° for the azimuth and elevation angle, correspondingly. The fiber orientation in the disordered dataset fluctuates from -90° to 90° for the azimuth angle and from 0° to 90° for elevation angle.

3.3. Implementation

The morphological analysis of fibrous structures is a composite task consisting of many intermediate steps and analysis strategies. Thus the proposed method [340] is only a small part of such analysis; therefore it was decided to develop quanfima (Quantitative Analysis of Fibrous Materials) package [341], which allows seamless integration of a complete fiber analysis into third-party software and data processing workflows based on the Python language. The package provides an automated analysis of the foremost parameters required for material characterization, such as porosity, fiber orientation and diameter, number of particles, and inclusions. It facilitates the 3D visualization of fibrous structures by mapping geo-coordinates to a color scheme, and the visualization of the orientation in a region-wise (*i.e.*, plotting orientation vectors in regions of interest distributed across the image) as well as a pixel-wise (*i.e.*, coloring pixels according to orientation in a neighborhood) fashion.

The quanfima package is based on several widespread third-party packages developed for data analysis, such as NumPy [342], SciPy [343], scikit-image [344], Numba [345], PyCUDA [346], TensorFlow [347], Pandas [348], and Statsmodels [349]. Their capabilities were combined to create a complete workflow for the characterization of structures. The functionality of quanfima is separated into four modules as shown in Fig. 3.6. In following, all externally imported libraries, packages or components will be called as modules to unify the architecture representation.

3. Orientation quantification method based on the ray-casting

The modules used by quanfima can be separated into mandatory and optional. The mandatory modules are depicted in white and ensure a core functionality of analysis routines and are therefore essential for installation. The optional modules drawn in gray and are required only by the visualization module and can be skipped if the user prefers to use other software for visualization. The dataset can be loaded as separate grayscale images, or as a stack of images representing a 3D volume. In the following, the modules will be described with emphasis on key functions.
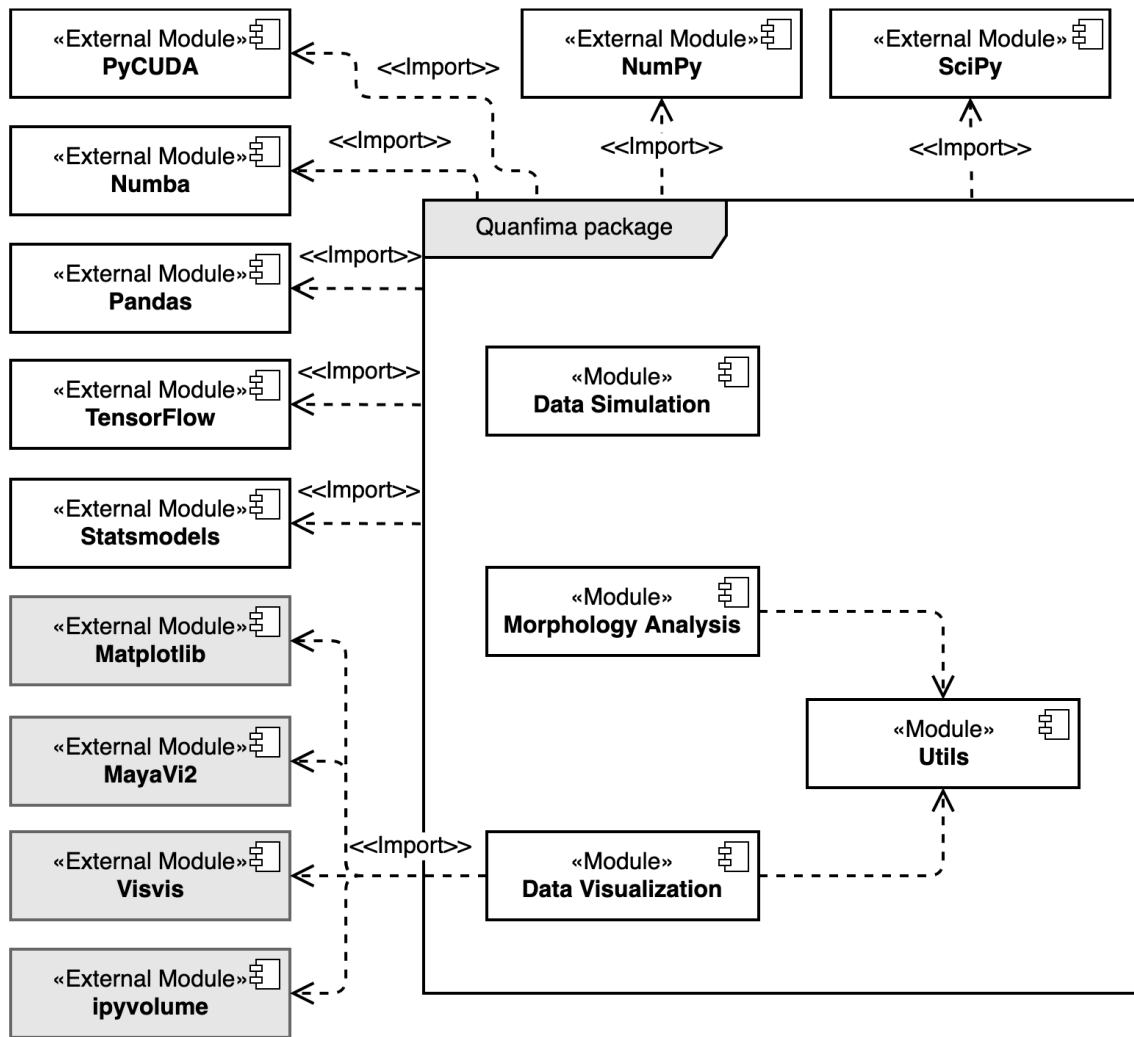
Figure 3.6.: The diagram of the quanfima package consisting of several modules to perform data simulation, analysis and visualization.

3.3.1. Simulation module

The simulation module in the quanfima package implements the algorithm described in Sec. 3.2.4, which allows generating synthetic datasets. Such datasets provide users with the ground-truth values which makes verification of imaging techniques or analysis algorithms possible. The simulation process is provided by `simulate_fibers` function, which takes a set of arguments: the size of the output simulated dataset, the number of fibers, the maximum number of attempts, and the ranges of radii, length, tilt angels and gap between fibers. The completion of the process occurs when the number of attempts to generate a fiber exceeds the specified number of maximum attempts. The module was used to generate datasets for the performance evaluation in Sec. 3.4.1.

3.3.2. Morphology module

This module is a core part of the developed package. It offers functionality for analysis of fibrous structures, which includes the estimation of diameter, orientation, and the number of fibers, as well as allows porosity calculations. The intermediate steps such as data pre-processing, segmentation and medial axis extraction are provided by third-party modules (*e.g.,* scikit-image) as shown in Fig. 3.7.
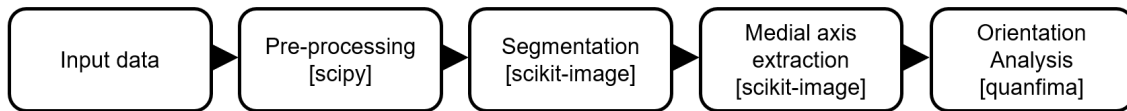

Figure 3.7.: The general workflow for fibrous orientation analysis with a specified software package for each stage.

Fiber orientation analysis

The fiber orientation analysis is performed on skeletonized binary data obtained via a segmentation procedure from the scikit-image module included in quanfima, or any other modules which provides image segmentation capabilities.

This module provides implementations of three functions for analysis of the fiber orientation, `orientation_fourier`, `orientation_3d_tensor`, and `orientation_3d_ray_cast`, which are based on the Fourier spectrum [350], the second-order structure tensor [351, 352, 235, 353], and the ray-casting approach proposed in Sec. 3.2.3. These functions take the input binary data of the suitable dimension and return a value or a pair of numbers representing the orientation angles in radians.

Fiber diameter estimation

The estimation of fiber diameters is provided by a function `estimate_diameter_single_run`. It requires a binary dataset obtained by segmentation and including orientation information at every point of the medial axis of the fibers, as obtained by the fiber orientation analysis. The function returns a dataset, where each voxel is represented by an average fiber diameter calculated within its neighborhood.

Porosity estimation

The porosity calculation is provided by the function `calc_porosity`, which returns an associative array, where keys are intensities, and values are the corresponding porosity estimations. The required input is a segmented dataset, which

is composed of an arbitrary number of materials N_m, labeled from 1 to N_m on a zero-background. The porosity is measured as the fraction of the air volume or material volume to the whole volume.

Object counting

Objects may be counted by the `object_counter` function. It allows for counting and the estimation of various morphometric characteristics of non-adjacent particles and inclusions presented in the segmented dataset. Behind the scene, it performs a connected-component analysis to assign different labels to the non-adjacent clusters of the voxels. Then, characteristics are calculated for each separated object, and the results are saved as a CSV file.

3.3.3. Visualization module

Visualization is a routine task for data analysis. In many cases, it requires exporting data to the format required by the visualization software and needs adjusting many parameters. To avoid such extensive work, this module was added. It allows visualizing all extracted quantitative characteristics like orientation and diameter via histograms, heatmaps, and volume rendering to reveal so far hidden properties. The module is largely based on available modules, such as Matplotlib [354], VisVis, Mayavi2 [355], and ipyvolume. If pre-processing is required before visualization, a set of functions provided in the utils module described in the following simplifies this procedure.

3.3.4. Utils module

The volumetric visualization of fiber orientation requires mapping of geo-coordinates to a specific color. The function `geo2rgb` juxtaposes a color from the HSV color model to a pair of azimuth and elevation values.

Statistical information

For statistical analysis, such as the calculation of p-values, the function `calculate_tukey_posthoc` was included. To estimate p-values for several sample characteristics, the user passes an associative array with values and their type, which store the array of sample characteristic values (*e.g.*, porosity) and the array of labels (types of biomaterials) associating each value with a specific a group of samples. The result of the p-value calculation may be saved to a CSV file.

3.4. Performance evaluation

The proposed method and the tensor-based approach were embedded into the described workflow to estimate accuracy and throughput of orientation quantification on a synthetic dataset in various conditions.

3.4.1. Dataset description

The synthetic dataset was generated as described in (Sec. 3.2.4) to validate the proposed method and compare it to the tensor-based approach. The structure of the dataset resembles a real-world fiber-reinforced material. It is composed of 70 fibers of radius from 3 to 20 pixels with a gap between them from 3 to 10 pixels, oriented in a range from -90° to 90° of azimuth and from 0° to 90° of elevation angle components. Afterward, it was contaminated with the additive Gaussian noise with different the standard deviation σ_{agn} of 0.5, 1.0, and 1.5, and subsequent smearing with the Gaussian filter with σ_{smooth} of 1.0 and 2.0.
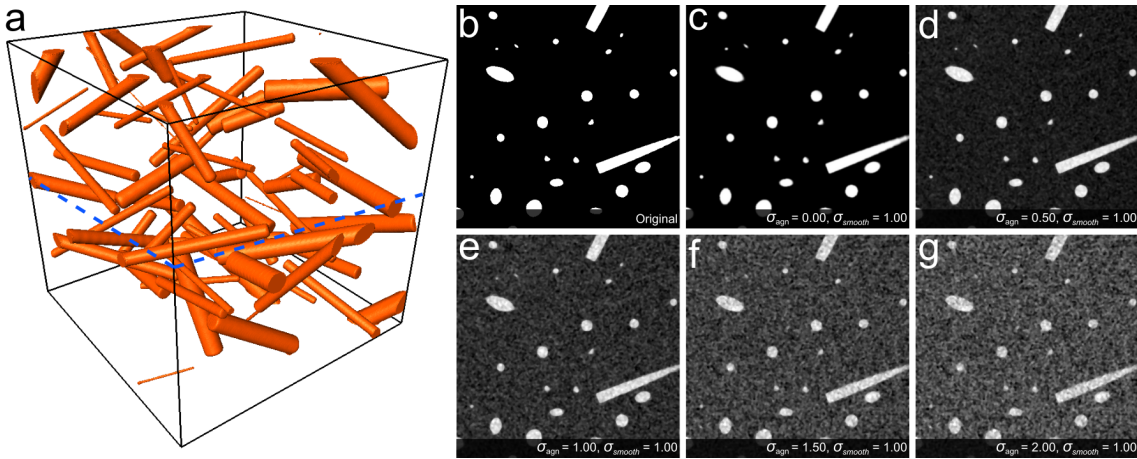
Figure 3.8.: The synthetic dataset (a) with the central slice extracted from the XY-plane (marked with the blue dashed line) of the same dataset contaminated with additive Gaussian noise and smeared with the Gaussian filter (b-g).

3.4.2. Validation procedure

The procedure is composed of several phases. At the first phase, the noiseless dataset is estimated by each method with varying a window size from 4 to 44 pixels, selected heuristically based on the radii of fibers in the synthetic dataset. Then, the window size providing the best accuracy is selected for further calculations. Thereafter, the fixed window size is used in the processing of the generated dataset to observe the behavior and limitations of each method in specific conditions. In the end, the throughput of the methods is estimated at different scales of the same dataset.

3.4.3. Analysis workflow

The implemented workflow was composed of the following stages: the pre-processing stage uses the non-local means filter in a slice-wise fashion with the smoothing parameter $\gamma_{sm} = 7$, the patch size $W_{tp} = 3$, and the size of the search window $W_{sw} = 21$, which were determined heuristically, the segmentation stage employs the Otsu thresholding algorithm [356] to obtain the binary data, the medial axis extraction was performed with a parallel thinning algorithm [337], and at the analysis stage both methods were estimated in the specified 3D local window. The entire workflow was implemented and performed on CPU, except the orientation analysis methods, which were implemented and executed on both CPU and GPU.

3.4.4. Benchmarking setup

The performance evaluation was done at a computer operating under 64-bit Ubuntu 16.04 and equipped with Intel Xeon E5-4660 v4 processor, NVIDIA Tesla T4 16GB graphical adapter and 60 GB of random access memory. The data were located at the Large Scale Data Facility [357] and were accessed via a high-speed network of 320 MB/s.

3.4.5. Orientation evaluation

The synthetic dataset was analyzed with each method for a range of window sizes from 4 to 44 pixels to determine the optimal one. The methods have similar behavior of elevation component errors as shown in Fig. 3.9(a,b), which rapidly fall from 4 to 20 pixels of window size and then do not change much. The results presented in Fig. 3.9a for the tensor-based method show that the sum of absolute angular errors of azimuth component rapidly falls from 4 to 28 and starts slowly decrease from 32 pixels of window size.

While the azimuthal error of the proposed method in Fig. 3.9b quickly falls from 4 to 24 and then slightly varies. Since the sums of absolute errors of both methods start slowly decreasing from a certain window size, the optimal window size is 34 pixels for the tensor-based approach and 32 pixels in a case of the proposed method was chosen. The optimal window sizes were then averaged to unify the parameters of both methods for the further analysis, which results in 33 pixels of window size. The absolute errors produced by the tensor-based approach and the proposed method using the optimal window size are (5.6°±24.29°, 1.03°±0.67°) and (3.75°±8.97°, 0.93°±1.25°) correspondingly, where the values in the parentheses are azimuth and elevation components.
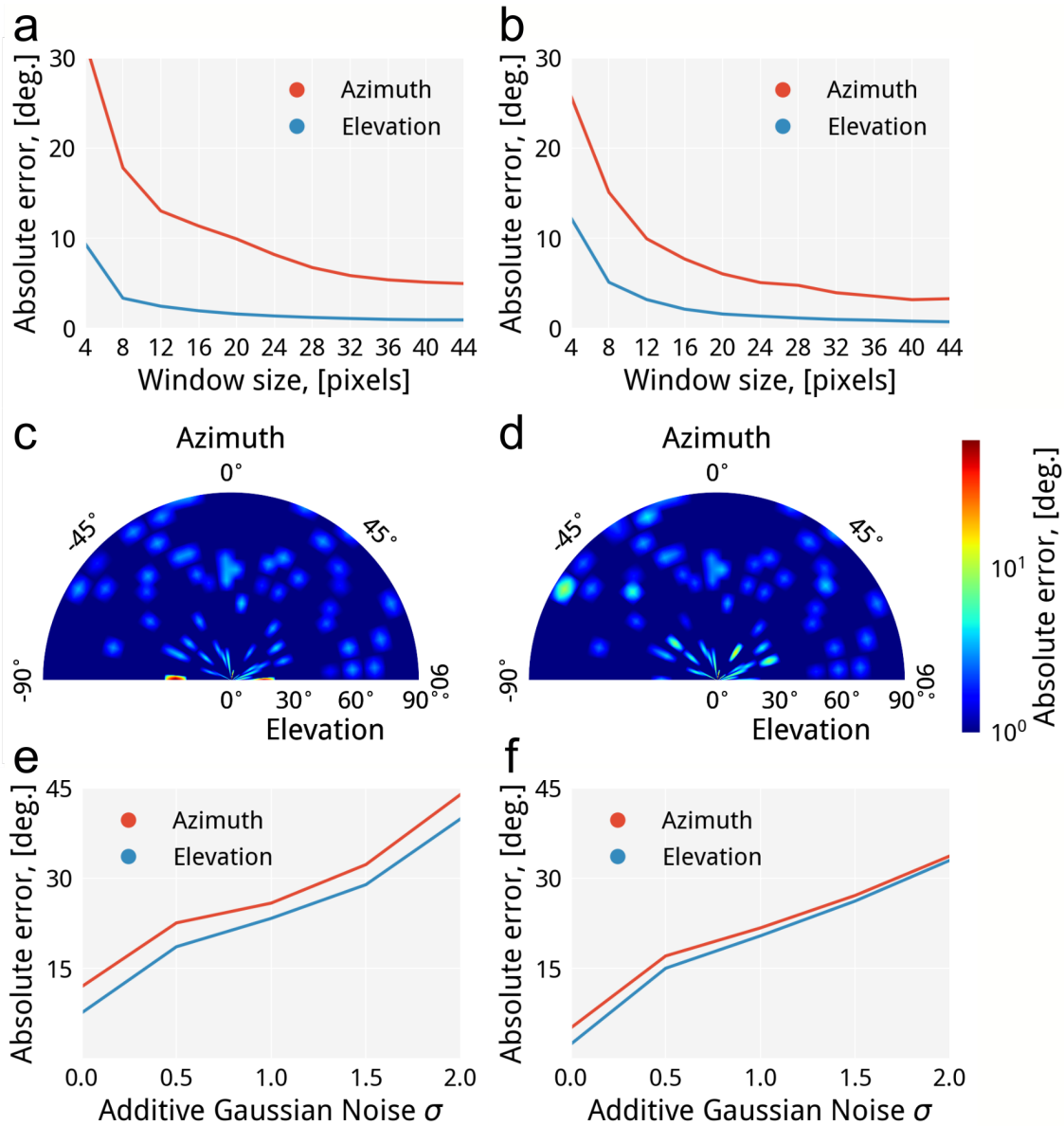
Figure 3.9.: The accuracy comparative analysis of the tensor-based approach (left) and the proposed method (right): a,b) the absolute error of orientation quantification of the noiseless dataset to determine the optimal window size; c,d) the evaluation of the sum of absolute errors to spot error-prone regions; e,f) the behavior of the absolute error while increasing of σ_{agn}.

Then, angular ranges were analyzed to find the most error-prone for each method while keeping the optimal value of the window size. The polar heatmaps depicted in Fig. 3.9(c,d) use a logarithmic scale to present the errors calculated as the sum of the averaged azimuth and elevation errors within angular ranges from -90° to 90° and from 0° to 90° for azimuth and elevation correspondingly, with a step of 5° for both. It can be seen from the Fig. 3.9c that most errors of the tensor-based approach are uniformly distributed over the heatmap, except the strong error peaks around (88°, 25°), (-88°, 18°) locations of azimuth and el-

evation components correspondingly. The proposed method has a comparable distribution of errors to its competitor and has weak error peaks around (-60°, 85°), (-50°, 26°), (25°, 16°) and (70°, 26°) locations of azimuth and elevation components correspondingly (Fig. 3.9d). The angular error increases towards angles 0°, 90°, and -90° because orientation at these angles tends to be incorrectly determined due to the limited spatial resolution governed by the window size, which in turn is restricted by the proximity of fibers. Thus, the optimal window size should maximize the spatial resolution and minimize capturing neighbor fibers in the window. In a case of tightly packed fibers, the optimal window size will be equal to the diameter of fibers, which is not sufficient to resolve a complete angular range.

Afterward, the methods were run over the synthetic dataset contaminated with σ_{agn} of 0.0, 0.5, 1.0, 1.5, 2.0 and σ_{smooth} of 1.0 to estimate the robustness (Fig. 3.8(b-g)). The results presented in Tab. 3.1 have shown that the proposed ray-casting approach is more accurate for both azimuth and elevation components for every configuration of noise (Fig. 3.8(e-f)). The angular error almost linearly changes for the proposed method from 0.5 to 2.0 of σ_{agn}. However, it non-linearly changes in the same range for the tensor-based approach. This is due to the fact that the segmentation stage produces the over-segmented binary data because of the imposed noise, and subsequently, the extracted skeleton will be greatly distorted, and many wrong locations of orientation estimation are produced.

σ_{agn}	Ray-casting approach (R)		Tensor-based approach (T)		Difference |T-R|	
	Azth (°)	Elev (°)	Azth (°)	Elev (°)	Azth (°)	Elev (°)
0.0	5.11	2.41	11.95	7.59	6.84	5.18
0.5	17.05	15.01	22.55	18.59	5.5	3.58
1.0	21.71	20.39	25.84	23.30	4.13	2.91
1.5	27.11	26.18	32.25	28.91	5.14	2.73
2.0	33.67	32.96	43.91	39.86	10.24	6.9

Table 3.1.: The accuracy of methods for the synthetic dataset contaminated with varying σ_{agn}.

This validation procedure has shown that the proposed method provides higher accuracy than the tensor-based approach for the same dataset over different validation scenarios.

3.4.6. Throughput evaluation

The complexity of each method was analytically estimated in terms of the necessary number of arithmetical operations. It showed that the proposed method requires approximately $\sqrt{(N_\phi \cdot N_\theta) \cdot 2 \cdot N_{ws}^2}$ arithmetic operations if it is assumed that each ray is fixed length of $\sqrt{2 \cdot N_{ws}^2}$ and $(N_\phi \cdot N_\theta)$ rays were emitted in all possible directions in the 3D local window of size N_{ws} pixels; thus the computational burden highly depends on the number of emitted rays. Whereas, the tensor-based approach requires $N_{ws}^2 \cdot (10/3 \cdot N_d^3 + N_d^2) + (6 \cdot N_d^2 + N_d)$ arithmetic operations, where N_d is the size of the square matrix, which is in our case the 3×3 covariance matrix calculated from the 3D local window. The number of required operations was numerically estimated by substituting the variables with corresponding values used in the study, where $N_{ws} = 33$, $N_d = 3$, $N_\phi = 90$ and $N_\theta = 180$. This showed that the proposed method requires approximately $30 \cdot 106$ operations, while the tensor-based method needs 10^5 operations. Therefore, the throughput of the sequential computation of the proposed method theoretically is more than an order of magnitude less than for the tensor-based approach. However, this issue can be overcome with the help of CPU-specific code vectorizing optimizations and GPUs which are aimed at massive parallelization of fine-grained tasks.

The performance of the proposed and the tensor-based approach was experimentally evaluated over the synthetic dataset at different scales. The versions of the algorithms were implemented for CPU and GPU to compare their throughput in conditions of varying the computational environment and the sizes of the dataset. The results of the evaluation are presented in Fig. 3.10 and Tab. 3.2.
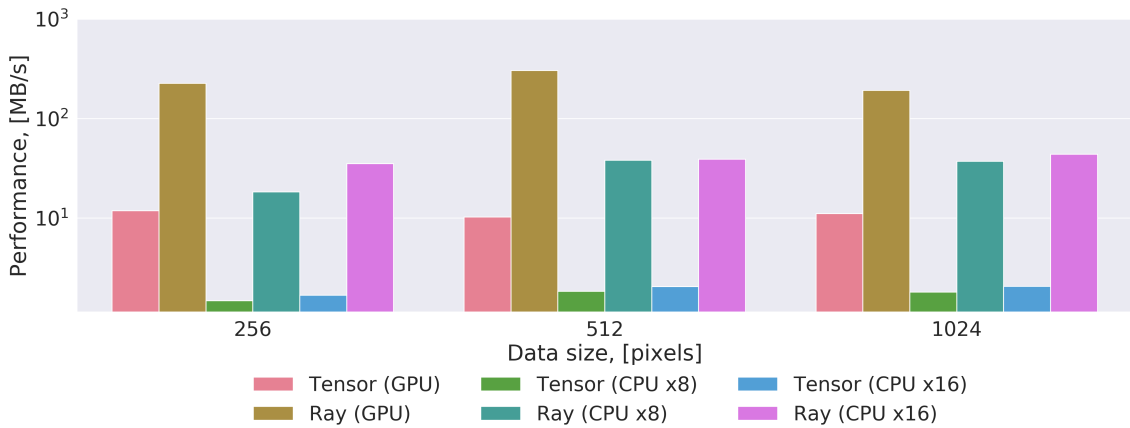
Figure 3.10.: The throughput evaluation of the proposed method (Ray) and the tensor-based approach (Tensor) over different data sizes for various computation environments.

The CPU version of the proposed method is 20 times faster than the tensor-based approach in the same conditions, and this tendency is preserved for all cases of computations involving CPUs. The throughput of the proposed method on CPUs is always higher than the tensor-based method because it is based on control flow

operators and trivial memory access patterns allowing for hardware-specific optimizations, whereas the method competitor is locked to the specific implementation of the matrix factorization algorithm. In the case of the GPU version, the proposed method outruns in 17-30 times the tensor-based approach, because the latter cannot be easily parallelized due to the matrix factorization algorithm. Thus, the undertaken experiment showed the high suitability of the proposed method for implementation on both CPU and GPU and its superiority over the tensor-based approach in all considered evaluation scenarios.

Data size (pixels)	Ray-casting approach (MB/s)			Tensor-based approach (MB/s)		
	GPU	CPU x8	CPU x16	GPU	CPU x8	CPU x16
256	**227.03**	**18.33**	**35.41**	11.90	1.47	1.68
512	**304.18**	**38.26**	**39.25**	10.25	1.84	2.05
1024	**192.66**	**37.34**	**43.92**	11.07	1.80	2.07

Table 3.2.: The results of throughput evaluation of the proposed method and the tensor-based approach for various data sizes and computation environments.

3.5. Summary

In recent years a vast number of methods aimed at orientation analysis of structures in datasets of diverse modalities were presented. Initially, every method was developed to answer a specific question about a particular dataset. However, late due to the inherent generality, some of them were successfully applied to other problems, such methods as analysis of the Fourier spectrum or the tensor-based approach. The latter has been compared to the proposed ray-casting method in this chapter.

The proposed method (Sec. 3.2.3) and the tensor-based approach (Sec. 3.2.2) were validated on the synthetic ground-truth dataset generated using the new algorithm presented in (Sec. 3.2.4). The results were shown that the proposed method surpasses the tensor-based approach in terms of accuracy, and due to inherent parallelizability, it can be efficiently implemented for GPUs to improve the execution time drastically. Moreover, the processing speed can even be further improved by shrinking the scanning angular ranges. The coarse-to-fine approach can be used to create the multi-level pyramid of the dataset. Then, starting from the top level, the orientation angles are calculated at each point and propagated to the next level. The orientation is recalculated for the new level at each point, taking into account the angular values at the previous level and some confidence in-

tervals to mitigate the accuracy errors. The process repeats until it reaches the latest level, where the initially large scanning ranges are shrunk to restricted ranges. The described changes will be introduced in the further revisions. Despite the proposed method outperformed the tensor-based approach, it cannot completely replace it. The method presented in this chapter is mainly oriented to the analysis of clearly separable structures because it relies on intensity accumulation along a ray path. While the method competitor is more suitable to analyze stuck together or bent structures by quantifying average orientations in regions-of-interests.

The analyzed methods were implemented as a part of the quanfima package for the Python language to perform a complete, comprehensive 2D and 3D analysis of fibrous structures. It provides capabilities for morphological and statistical analysis, and visualization without bindings to a specific visualization system. The package can be easily applied to fibrous structures of various nature for characterization of wall thicknesses, inclusions, and porosity. The power of such analyses will be demonstrated in the next chapter on analysis of polycaprolactone 3D scaffolds. In future work, the functionality of the quanfima package will be extended by implementing new algorithms to analyze fiber intersection types, fiber clustering, and fiber tracking providing the information about individual fiber lengths. It is also planned to improve the synthetic fiber generation algorithm which will no more be limited to straight fibers but allows for curved fibers with inclusions resembling particles and cells on a fiber surface.

4. Applications

4.1. Quantitative morphometric analysis of adult teleosts

A genotype-phenotype association plays an essential role in many studies, *e.g.*, etiology of diseases. Such associations can be readily found for species with a natural tissue transparency to visible light. These properties are inherent to small teleost models such as medaka (*Oryzias latipes*) and zebrafish (*Danio rerio*) [358–360]. Their embryos are small and transparent that allows for high-resolution analysis using confocal and wide-field light microscopy. However, detailed analysis of adult morphometrics is problematic since tissue thickness and pigmentation often impede analysis based on visible light. Moreover, quantitative analysis of traits requires detailed measurements of the entire body at high resolution. Due to the high demand for using these genetic models in studying diseases, it is crucial to provide imaging and data analysis approaches to permit the whole-body morphometric characterization of adult individuals.

The previous quantitative studies of morphometrics relied on landmark-based approaches which split traits into different groups [361] by linear measurements; however, they are limited because they do not take into account whole-body anatomical information. Recently, the potential of 3D imaging for quantitative estimation of phenotypes was demonstrated with optical projection tomography, µCT, and high-resolution episcopic microscopy of mice [362]. However, µCT technique provides the best imaging properties among other methods regarding penetration depth, attainable spatial resolution up to nanometres and scanning time. Similarly to visible light, it provides multiple imaging contrasts, such as absorption, phase and dark-field contrasts [363, 364]. The poor X-ray absorption of biological soft tissues can be circumvented with the X-ray phase-contrast technique operating at lower resolution [365, 366]. However, in some cases, absorption of X-rays can be enhanced by applying specific chemical contrast agents during a sample preparation stage, whereas phase-contrast requires to modify the X-ray optical system to reach a similar result.

Recently, a few morphometric studies using µCT have been presented for rainbow trout and zebrafish [367–369], which relied on manually or unsegmented data due to low differential contrast of soft tissues and spatial resolution. In the last years, advances in image processing for ultrasonic, MRI, X-ray, and

other imaging techniques allowed to replace tedious and time-consuming manual segmentation of specific organs with semi-automated and automated methods [370, 371]. However, automated whole-body segmentation still is not widely used due to its complexity, and so far among model organisms, it was successfully performed for mice [74, 372].

In this section, the application of the proposed segmentation framework (Sec. 2.1) to μCT data of small teleost medaka was presented (Fig. 4.1). The imaging setup and sample preparation stage were optimized for the specific spatial resolution and tissue contrast in a way that be applicable to large numbers of specimens required for comparative morphometric analysis. Thus, the whole procedure provides a general tool for high-resolution anatomical studies for small vertebrates and potentially can be used for discovering genotype-phenotype associations and complex genetic traits.
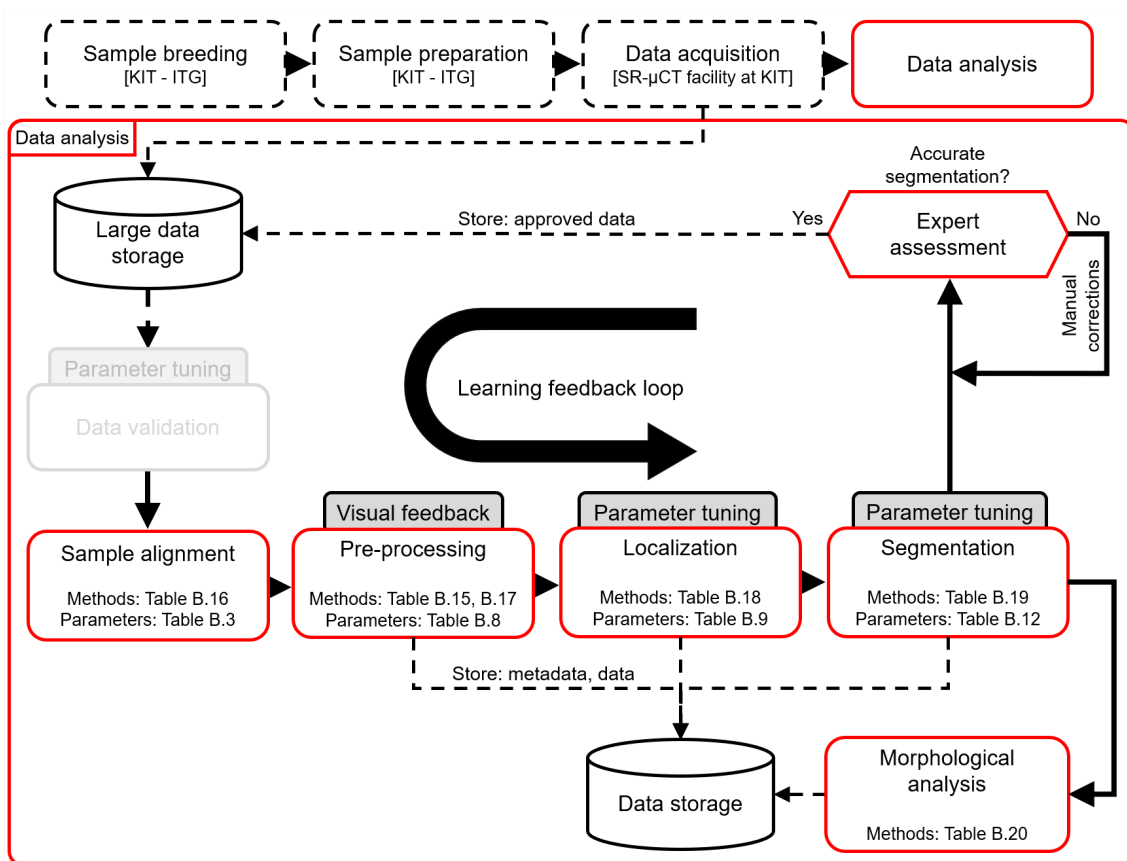
Figure 4.1.: The general diagram of the segmentation framework with the specific implementation and parameterization of each stage for μCT datasets of medaka fish. The data analysis stages are described with methods and parameters, while the data preparation stages are characterized by institutions and facilities where experiments were carried out (*e.g.*, the Institute of Toxicology and Genetics (ITG) of the Karlsruhe Institute of Technology (KIT)). The unused stages are marked in gray.

4.1.1. Data segmentation and analysis

The reconstructed and manually segmented datasets were split into 10 training and 10 validation datasets. Then, they were processed and analyzed with the segmentation framework presented in (Sec. 2.1) to be prepared for morphometric analysis of the brain, liver, head nephrons, and heart organs between different inbred lines. In the beginning, the datasets were aligned to the reference dataset, and then they were pre-processed to prepare data for all further stages. The prepared data was then passed to the localization stage, where the specified organs were localized by determining their 3D bounding boxes. Next, the extracted bounding boxes were used in the segmentation stage to obtain binary masks of the specific organs, whereas the organs extracted from the manually segmented datasets were employed in the training procedure. Finally, the segmented organs can be subjected to morphometric analysis to determine their shape properties.

Alignment

The reference dataset was manually aligned so that the specimen was placed in the center and its sides corresponded to the orthogonal planes. The registration method utilizing the multi-scale approach and the rigid transformation model was parameterized as listed in Tab. B.3. It was initialized with four zoom levels with corresponding scale factors of 8, 4, 2 and 1, where at each zoom level, data is smoothed with the Gaussian filter of σ_{smooth} of 3, 2, 1 and 0, correspondingly. The optimization procedure employing the gradient descent method was initialized with the mutual information cost function (Eq. (2.12)), and $\epsilon = 0.0001$ was run over 1000, 1000, 1000, and 500 iterations for the corresponding zoom levels. The reference dataset was used as the fixed image and non-aligned datasets were used as the moving images in the context of the cost function. If the optimization process was converged before reaching the limit of iterations, then it stopped and went to the next level. The datasets considered for analysis were aligned to the reference dataset and showed relatively small errors. The results of alignment for the validation datasets are presented in Fig. 4.2.
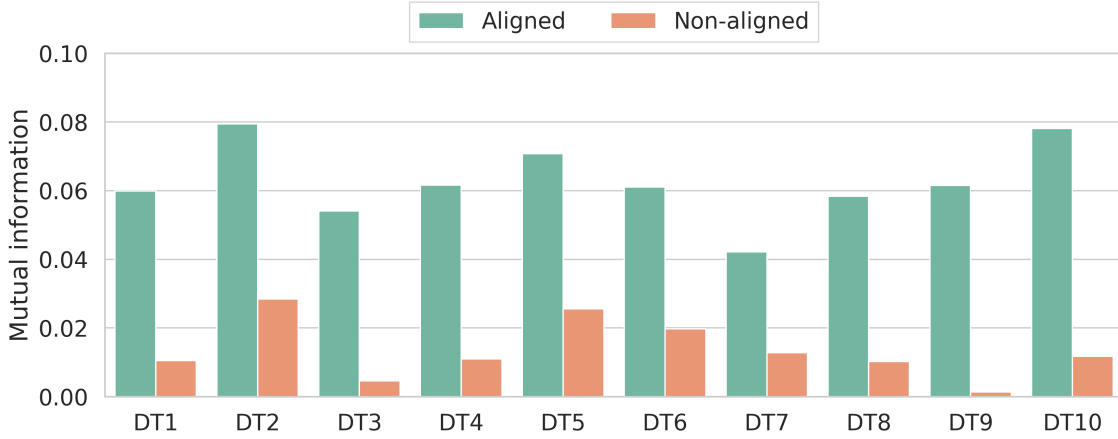
Figure 4.2.: The accuracy results of aligning medaka datasets calculated in terms of MI metric.

Pre-processing

The aligned datasets were then passed through the pre-processing stage configured as in Tab. B.8 to prepare data for the further workflow stages. The suitable denoising parameters, namely, the smoothing factor $\gamma_{sm} = 7$, the patch size $W_{tp} = 3$, and the search window $W_{sw} = 21$ were selected by visual inspection of results collected from the output of the denoising method over the various combinations of parameters. The parameters of morphological operations involved in the extraction of specimens such as the size of a structural element $S_{se} = 5$ were selected based on the smallest observable structure, whereas the number of iterations $N_{mi} = 3$ was chosen empirically. Then, the extracted specimens were spatially normalized to S_{norm} of 1152×1024×6656 voxels determined as the nearest size, which is divisible by two. Finally, the multi-scale pyramid of $N_L = 6$ levels, allowing to resolve the target organs at the last level was created from each normalized dataset. Every successive level of the pyramid starting from the first one is produced by downscaling data by $\gamma_{ds} = 2.0$ and smoothing with the Gaussian filter of $\sigma_{pyr} = 1.0$.

Localization

The pre-processed datasets were transferred to the localization stage, where a 3D bounding box was determined for each target organ.

The optimal architecture of CNN depends on the nature of data; therefore the networks were trained on several combinations of parameters listed in Tab. B.9 to determine the best-suited architecture. The slice configuration was built for $N_c = 3$ target organs from $N_{ns} = 3$ neighbor slices with a step of $N_{ss} = 1$ slice. The networks were parameterized with the number of base filters $N_{bf} = 16$, the minimal size of the receptive field $N_{mrf} = \{2, 4, 8\}$, the dropout ratio of $N_{dp} =$

0.5, and the number of neurons $N_{fc} = 1000$. The Adam optimization method with the learning rate $\alpha = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$ was used. Every network was trained on the datasets from the fourth level of the multi-scale pyramid, prepared along the corresponding orthogonal axis with the given slice configuration as described in Sec. 2.5.2. The number of training datasets was limited; therefore, the rigid and affine augmentation strategies parameterized as in Tab. B.10 were integrated into the training process, which was performed over 250 epochs with a batch size of 8 contextual slices. The learning rate was reduced by a factor of 2 during the training if a loss function was not improved for 5 epochs. If the accuracy on the validation datasets stopped improving for last 10 epochs, the training was stopped. Thus, the network parameters providing the smallest wall-distance error (Eq. (2.47)) on the validation datasets was selected.

The input data prepared for the specified orthogonal axis as described in (Sec. 2.5.2) was passed into the corresponding trained CNN, which predicted the walls of the bounding box for each target organ along the corresponding axis. Then, the obtained predictions from all orthogonal axes were converted into 3D bounding boxes using the method from (Sec. 2.5.5). Afterward, these 3D bounding boxes were propagated to the higher level of the multi-scale pyramid as described in (Sec. 2.5.6), where the profile thresholding value $\gamma_{th} = 0.1$ and the smallest size of the connected-component $N_{ml} = 64$.

The optimal network architecture for localization of medaka organs was found with the grid search method over the minimal receptive field size parameter, while others were fixed. The evaluation results presented in Fig. 4.3 demonstrate the average wall-distance error calculated across all the validation datasets for all the organ bounding boxes of medaka fish.
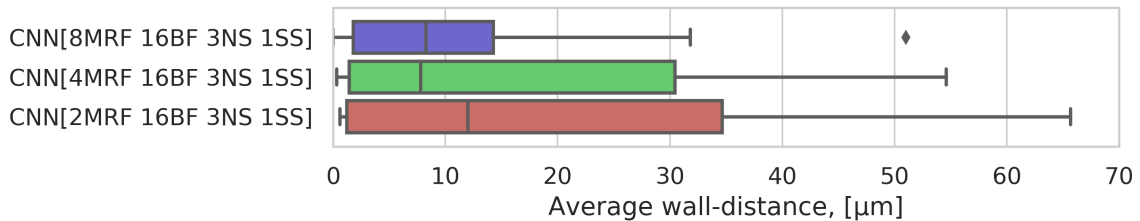
Figure 4.3.: The evaluation of the localization network architectures to determine the one providing the smallest average wall-distance error calculated over all the validation datasets across all the organ bounding boxes of medaka fish.

The network architecture `CNN[8MRF 16BF 3NS 1SS]` has the smallest wall-distance errors among other architectures as shown in Tab. B.11. This architecture has a less error differ by 5.22 µm and 8.53 µm for `CNN[4MRF 16BF 3NS 1SS]` and `CNN[2MRF 16BF 3NS 1SS]`, respectively.

The localization results of the brain, liver, head nephrons, and heart with different network configurations are presented in Fig. 4.4 and listed in Tab. 4.1, which

demonstrate that `CNN+R` provides the smallest wall-distance error for all organs, except the liver, among the other network configurations.

Network configuration	Organ name	Wall-distance (µm)
CNN	Brain	17.94±25.58
	Liver	18.96±22.89
	Head nephrons	27.43±28.9
	Heart	26.83±30.79
CNN+R	Brain	**2.53±1.99**
	Liver	4.83±3.96
	Head nephrons	**4.57±3.17**
	Heart	**5.31±3.47**
CNN+A	Brain	6.31±7.52
	Liver	**3.90±2.06**
	Head nephrons	7.65±6.89
	Heart	8.86±6.42

Table 4.1.: The organ localization results of medaka fish for various network configurations: no augmentation strategy (CNN), the rigid augmentation strategy (CNN+R), and the affine augmentation strategy (CNN+A).

The high values of standard deviation can be explained by the fact that the organs are poorly distinguishable because they are surrounded by tissues and other organs, in some cases, with a similar texture. Also, some datasets could not properly be aligned due to large anatomical or spatial location differences of specimens.

Thus, the network `CNN+R` is more accurate than `CNN` and `CNN+A` for the brain, the head nephrons, and the heart, however, it is inferior to `CNN+A` in the wall-distance of the liver. Despite its little weakness in localization of the liver, `CNN+R` was selected as the optimal network configuration, which can robustly localize the brain, liver, head nephrons, and heart. Its superiority is based on the usage of the rigid augmentation strategy, which allows correctly simulating the real geometric transformations which the sample undergoes. Whereas the network `CNN+A` uses the affine augmentation strategy taking into account more complex geometrical transformations, which do not exist in the real experiments and consequently lead to lower accuracy.
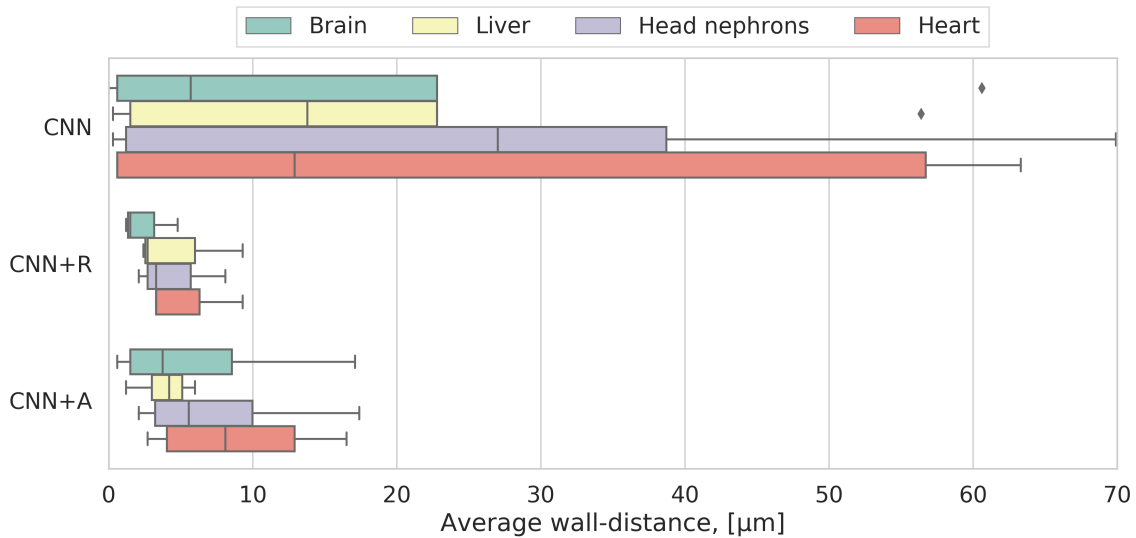
Figure 4.4.: The localization results of several medaka organs using the optimal network configuration and different augmentation strategies, which are calculated as the average wall-distance error for the specific organ across all the validation datasets.

Segmentation

The localized organs were then spatially normalized as described in (Sec. 2.6.2) and passed into the segmentation stage, where the pre-trained U-Nets performed the organ-wise segmentation for each organ individually.

The architecture of U-Net as well depends on data; therefore it was trained in a slice-wise fashion on various combinations of parameters listed in Tab. B.12 to determine the most efficient architecture for the specific organ. The networks were initialized with the number of base filters $N_{bf} = 32$ and the minimal size of the receptive field $N_{mrf} = \{8, 16, 32\}$, thus varying only a depth of the networks. The learning process was performed with the Adam optimization method parameterized with the learning rate $\alpha = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$. The networks were trained on slices obtained from the extracted datasets of the target organ along the corresponding orthogonal axes as described in (Sec. 2.6.2). Due to a lack of training datasets, similarly to the localization stage, the rigid and affine augmentation strategies (Tab. B.10) were employed in the training process, which lasted for 250 epochs with a batch size of 8 slices. The learning rate reduction and the training stop policies were used the same as in Sec. 4.1.1. Finally, the network architecture providing the highest Dice ratio, see Eq. (2.46) on the validation datasets was selected as the optimal architecture.

The U-Net predicts the presence of the target organ at every pixel of the input slice producing the output probability map of the same size as presented in Fig. 4.5. The slices along the specified orthogonal axis were prepared as explained in Sec. 2.6.2. The obtained probability maps along all axes were merged

and post-processed according to the procedure described in Sec. 2.6.6, and parameterized by the probability thresholding value of $\gamma_{seg} = 0.5$ to generate a 3D binary segmentation.
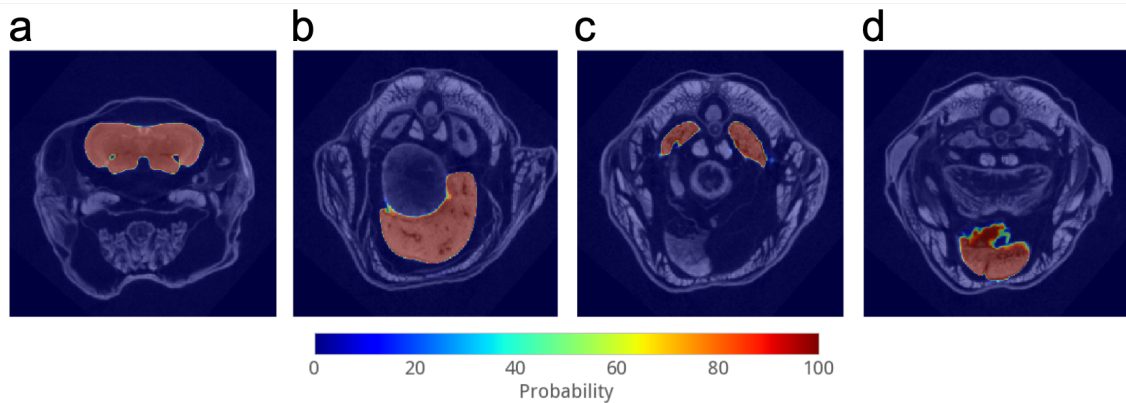

Figure 4.5.: The predicted probability maps overlaid on top of the corresponding axial slices of a medaka fish dataset for a brain (a), a liver (b), head nephrons (c) and a heart (d). The colorbar denotes the probability of presence of the specific structure at every pixel.

The optimal network architecture was determined with the grid search method over the minimal receptive field size, while fixing the number of base filters. The architecture evaluation results are presented in Fig. 4.6.
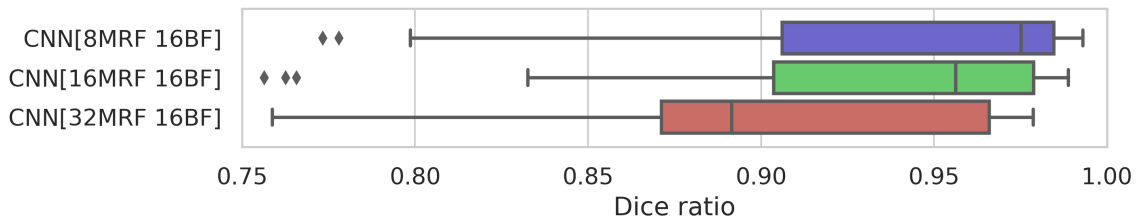
Figure 4.6.: The evaluation of the segmentation network architectures to determine the optimal one, providing the largest average Dice ratio calculated over all the validation datasets across all the segmented organs of medaka fish.

As it can be seen, the architecture `CNN[8MRF 16BF]` provides the segmentation accuracy of 0.92±0.08 whereas `CNN[16MRF 16BF]` and `CNN[32MRF 16BF]` have the accuracy of 0.91±0.07 and 0.89±0.08, respectively. Thus, the architecture `CNN[8MRF 16BF]` is superior to `CNN[16MRF 16BF]` and `CNN[32MRF 16BF]` by 0.01 and 0.03, correspondingly.

In the next step, the proper augmentation strategy was determined for the optimal network architecture `CNN[8MRF 16BF]` by evaluating various network configurations as in Sec. 2.9.7. The segmentation results of each organ produced by each network configuration are shown in Fig. 4.7, and additionally listed in

Tab. 4.2, which demonstrated that the network CNN+A provides the best segmentation accuracy for all organs.

Network configuration	Organ name	Dice ratio
CNN	Brain	0.97±0.01
	Liver	0.93±0.08
	Head nephrons	0.87±0.14
	Heart	0.89±0.11
CNN+R	Brain	0.97±0.01
	Liver	0.93±0.07
	Head nephrons	0.86±0.15
	Heart	0.85±0.13
CNN+A	Brain	**0.98±0.01**
	Liver	**0.94±0.06**
	Head nephrons	**0.88±0.11**
	Heart	**0.91±0.01**

Table 4.2.: The organ segmentation results of medaka fish for various network configuration.

As it can be seen, the network CNN+A is superior to CNN and CNN+R network configurations in the segmentation accuracy of every organ.
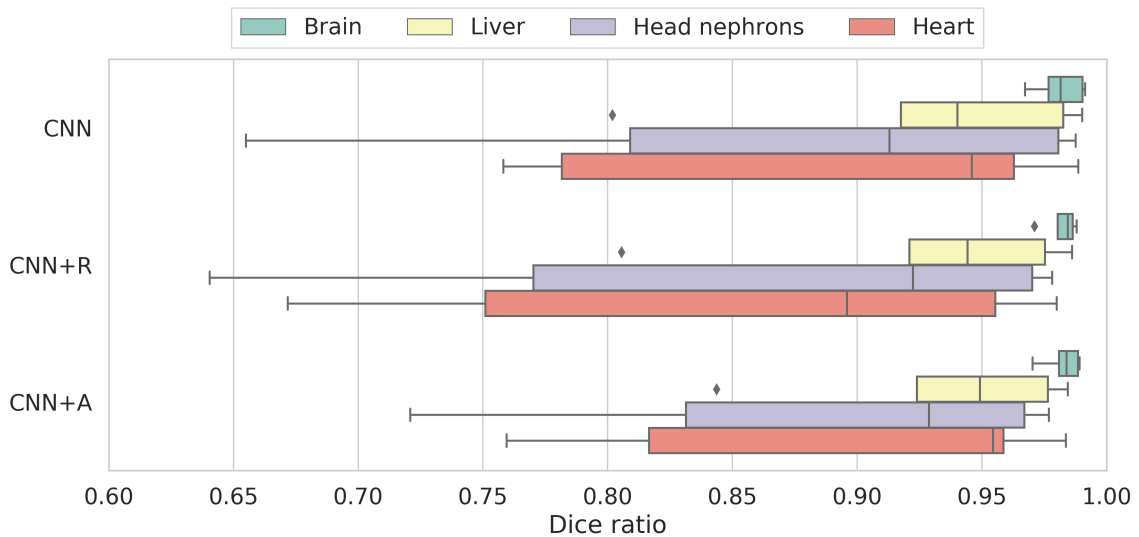

Figure 4.7.: The segmentation results of several medaka organs using the optimal network architecture and different augmentation strategies, which are calculated as the average Dice ratio for the specific organ across all the validation datasets.

The segmentation result of a new dataset is reviewed by the domain experts and

then added to the set of training datasets, which is then used to re-train the organ-specific U-Nets.

Discussion

The imaging setup used in the experiment was equipped with an automated robotic sample changer allowed for sample positioning with high precision. Therefore, the validation stage was skipped, and all datasets have proceeded directly to the alignment stage.

The automated mounting procedure eliminates the need to manually position a sample container, and subsequently reduces the risk of placing a sample out the field of view. Since all samples were already positioned approximately at the same location by the automated sample changer, it dramatically simplifies the registration problem by reducing the number of possible solutions. Therefore, alignment was done in a small number of iterations, since minor transformations were required to align all specimens.

The datasets mainly suffered from the statistical noise, which could interfere in the successful execution of the further stages. Therefore, the denoising step was necessary; moreover, the denoising parameters had to be chosen carefully by visual inspection of outputs produced with different settings. Since, the non-local means denoising algorithm is susceptible to parameterization, and can easily filter out essential features of the organs along with the noise.

The localization procedure was performed on highly downscaled data (6x) that significantly reduced the training and prediction time. The obtained bounding boxes were then upscaled and successfully applied in the extraction of the corresponding organs from data of higher resolution (4x). Despite that localization was done on a quite coarse resolution, most of the organs still were accurately localized. The organs were localized using the CNN with the optimal configuration of network architecture, which was determined by the grid search method over the minimal size of the receptive field parameter. Other parameters were fixed because they do not influence much the accuracy as it was shown in (Sec. 2.9.6). The lack of theoretical basis behind the deep learning does not allow inferring the optimal network configuration beforehand. Therefore, the only way to find the best-suited configuration for datasets of a particular nature, it is through a hyper-parameter optimization. However, there were problems with the head nephrons and heart, because their shape tends to vary among individuals even within the same inbred line significantly. While, the brain and liver were localized more robustly since their shape usually quite stable, and they occupy more volume than the head nephrons and heart that allows them to be more perceivable. The localization inaccuracies are also explainable by a lack of training data despite the usage of the rigid and affine augmentation strategies, which seems to be cannot wholly simulate plausible organ shape variations. The piece-wise augmentation strategy was not used, because it provides only random local deformations,

which might be unrelated to the real behavior of anatomy. It would lead to training the network for recognizing anatomical deformations, which do not exist in the real organs and subsequent failure in real datasets. The anatomy-driven data augmentation should be introduced, which would take into account plausible anatomical variations to cope with this issue.

The obtained bounding boxes were used to extract the organs of interest, such as the brain, liver, head nephrons, and heart. The optimal network architecture was determined using the same approach as in the case of localization, by performing the grid search over the minimal size of the receptive field parameter, which regulates the depth of network. Similar to the localization stage, a lack of data was compensated by incorporating the rigid and affine augmentation strategies into the training process. Nevertheless, this stage shares the same issues regarding the augmentation process with the localization stage, because in both cases, augmentation may introduce unrealistic shape variations into the prediction model that in turn would degrade prediction accuracy on the real datasets.

Thus, a lack of training data is the current bottleneck for localization and segmentation stages. It can be partially mitigated by introducing more manually segmented data and integrating information about anatomy variations into the augmentation process.

4.1.2. Materials and methods

Fish stocks and husbandry

The stock of Medaka (*Oryzias latipes*) was maintained at ITG of KIT at 26° C for 14 hours of light and 10 hours of dark conditions. All experimental procedures and animal husbandry were done according to local and European Union animal welfare standards (Tierschutzgesetz 111, Abs. 1, Nr. 1, AZ35-9185.64/BH).

Optimisation of absorption contrast for μCT of teleosts

The μCT allows performing 3D imaging for various applications, and it provides high penetration power, different spatial resolution scales, and multiple contrast modes. To acquire high-resolution data of adult teleosts possessing the highest possible contrast, both between and within various organs, and facilitating automated segmentation, various staining procedures to optimize absorption contrast were studied. The previously proposed staining protocols were optimized for overall, rather than differential contrast, or were designed to ensure contrast only for some specific organ or tissue. Also, these studies were mainly oriented to embryonic or even earlier developmental stages of specimens. Several fixation and contrasting protocols were systematically evaluated in [373] to establish

an optimal procedure providing sufficient differential contrast between individual organs and sub-regions within organs, enabling high-resolution analysis of whole adult specimens.

Various contrast agents were evaluated, such as, iodine (I_2KI), phosphotungstic acid (PTA), and europium chloride ($EuCl_3$). The results of the preparation of whole adult medaka with each contrast agent are shown in Fig. 4.8 on the representative coronal and sagittal cross-sections through the center of a 3D volume. As a result, a new sample preparation protocol based on a PTA contrast agent was developed [373], which showed high and well-balanced differential contrast for all tissues, providing optimal contrast for µCT imaging of adult teleosts.
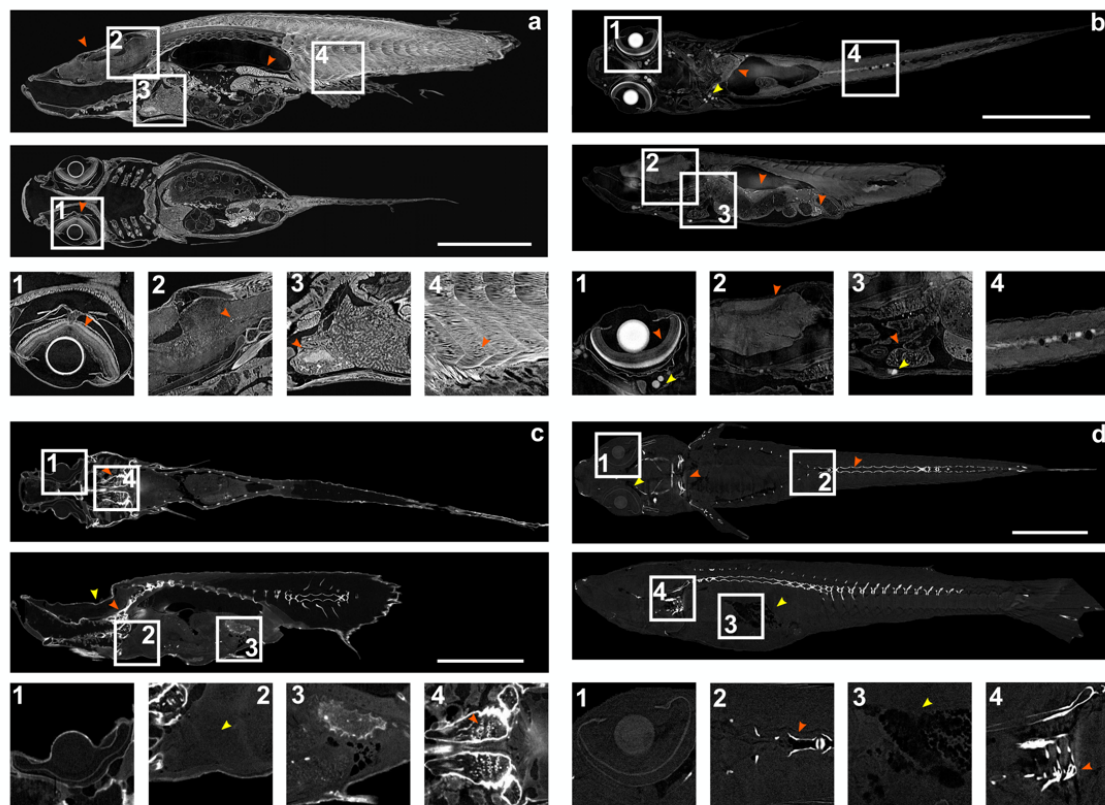
Figure 4.8.: The optimization of absorption contrast. The coronal and sagittal slices of adult medaka stained with PTA, I_2KI, $EuCl_3$ and unstained control. (a) PTA staining provides good differential absorption contrast in all body regions (red arrowheads). (b) I_2KI staining results in differential absorption contrast in the eye (1, red arrowhead). Its precipitates lead to agglomerates of high intensity concentrated in the head region, intestine, and spine (1, 3, yellow arrowheads; 4, spine). Absorption in inner organs is weak with low absorbing tissue (2, 3, red arrowheads). (c) $EuCl_3$ staining leads to strong absorption in bones (red arrowheads) and weak absorption in inner organs (2, yellow arrowhead). (d) In unstained medaka adults, the eye (1) and gut (3, yellow arrowhead) have very weak absorption, whereas bones possess strong absorption (2, 4, red arrowheads). The details are described in [373]. Scale bars: 6 mm.

The visualization of PTA stained adult medaka presented in Fig. 4.9, which proves that PTA allows revealing detailed structural information. The entire rendering (Fig. 4.9b) shows high differential contrast and high spatial resolution, which allows distinguishing even tiny details and different structures within organs. As examples, visualizations of the brain, eye, and intestine were shown in Fig. 4.9(a,c,d), demonstrating a potential for revealing detailed tissue morphology.
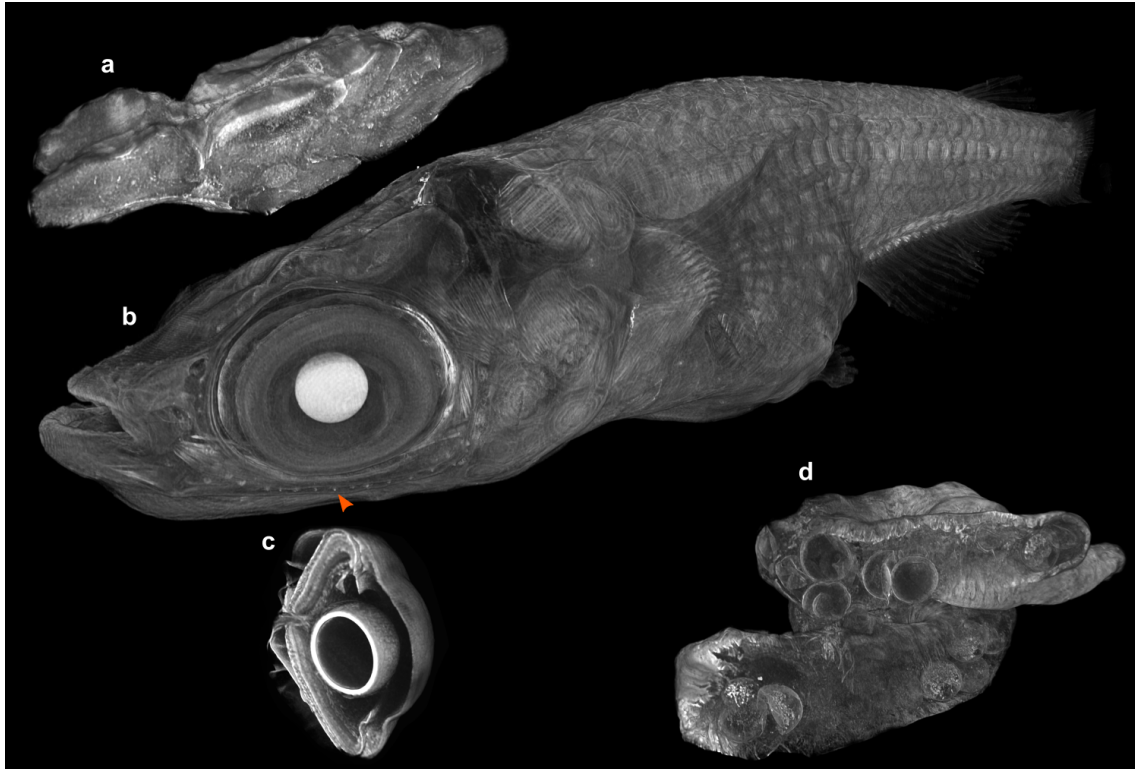
Figure 4.9.: The 3D renderings of the PTA stained adult medaka: a) the brain at 5x magnification; b) the entire body, length is approximately 22 mm (red arrowhead indicates neuromast of the cranial lateral line); c) the eye with transversal sectioning at 10x magnification; d) the intestine at 3x magnification (the spherical objects within the gut are artemia eggs).

This protocol was successfully integrated into the imaging pipeline, which resulted in high differential absorption contrast of soft tissues of adult specimens while preserving high spatial resolution required for automated segmentation.

Sample preparation

The 8 weeks post-hatching (sexually mature) medaka fish were sacrificed and fixed with 0.3% phosphotungstic acid in 70% ethanol. The samples were fixed for three days at room temperature followed by staining with a contrast agent for three days at room temperature, while the solutions were changed every 48

hours. After the staining procedure, samples were washed in PBS and embedded in 4% agarose and placed in polypropylene containers. The whole procedure with all other contrast agents is described in [373].

Data generation pipeline

The pipeline of the whole process is presented in Fig. 4.10, which illustrates all steps from the sample preparation to the data analysis.
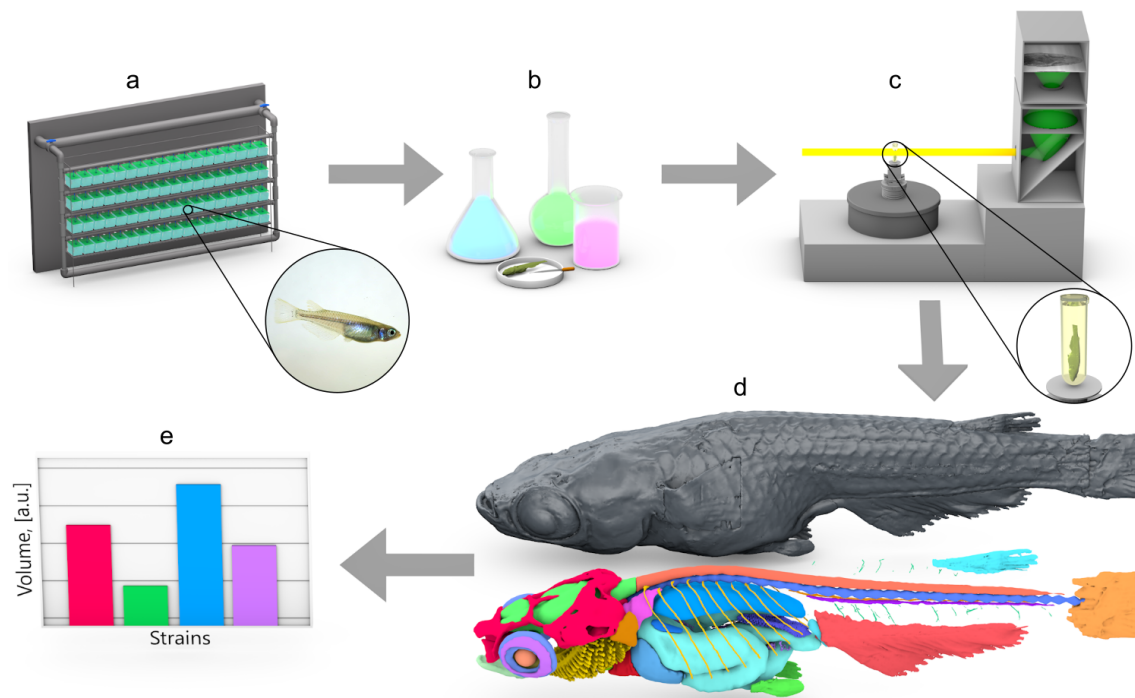
Figure 4.10.: The pipeline for quantitative 3D morphometric analysis of teleosts: a) fish husbandry; b) sample preparation by fixating and staining with a contrast agent; c) µCT data acquisition at the synchrotron radiation facility; d) tomographic reconstruction and manual segmentation of the training data for CNNs; e) automated segmentation and morphometric analysis.

The samples were scanned at the high-resolution tomography station of the synchrotron light source facility at KIT [374, 375] using a white beam from a bending magnet. The X-rays were filtered to 16 keV with a double-multilayer monochromator with an additional Al filter (0.2 mm). The X-ray projections were detected by an sCMOS camera (2560×2160 pixels, 6.5×6.5 µm^2 pixel size) coupled with an optical light microscope of 3.6x magnification. The X-ray beam was converted into visible light by a $Lu_3Al_5O_{12}$ (LAG) scintillator of 50 µm. The whole optical system resulted in an effective pixel size of 1.8 µm and the measured spatial resolution of 5 µm. From every sample, 2000 projections were acquired over 360° rotation. Since the samples could not fit the field of view, they were vertically

translated after each scanning session, and the obtained projections were then vertically concatenated to form a complete sample.

The acquired projections were corrected for dark noise of the camera, instability of the beam and inhomogeneities of the scintillator. Afterward, the 3D dataset was reconstructed from the corresponding set of corrected projections using the FBP algorithm provided by a GPU-based data processing framework [376].

Training data preparation

The brain, liver, heart, kidney and spleen organs were manually segmented for 10 datasets with low precision at each twentieth slice and then were fine-tuned and interpolated through the volume with the biomedical image segmentation application [377].

4.1.3. Conclusion

The quantitative morphometric analysis is crucial in studying genotype-phenotype associations. The μCT imaging provides unique characteristics, such as high penetration power, fast speed and applicability to samples presented in different resolution scales [378]. The chemical contrast agents allow increasing absorption of X-rays for biological soft tissues, and thereby they can be visualized [32].

In this section, the application of μCT for high-resolution imaging of the adult medaka was demonstrated. It provided a complete description of internal structures for quantitative morphometric analysis. The whole pipeline consisting of sample preparation, data acquisition, and data analysis using the framework presented in (Sec. 2.1) was implemented. The pipeline was evaluated on medaka fish, which is widely used as a genetic teleost model. PTA contrast agent providing high differential contrast in conjunction with high-resolution μCT imaging allowed for whole-body imaging while maintaining differentiation between tissues and organs. The automated segmentation enables extraction of morphometric parameters of internal organs that proves the applicability of the proposed segmentation framework for such kind of experiments.

The described pipeline for analysis of small adult teleosts provides several advantages. It allows for the entire body imaging while maintaining resolvability of all organs and tissues, so they can be perceivable for segmentation algorithms to perform the morphometrical analysis. The proposed framework reduced a computational burden by employing hierarchical analysis, thus enabling to process and analyze separate organs in large datasets. It allows uncovering genotype-phenotype associations, such as genome-wide association studies (GWAS). Thus, the proposed pipeline paves the way for state-of-the-art genomic approaches to

unravel the genetics underlying variance of quantitative traits for various organs.

Moreover, the developments in real-time tomographic reconstruction and rapid data processing algorithms, monitoring and controlling efficiency of imaging experiments employing µCT will allow for upscaling the whole pipeline to manage a large number of individuals. The small vertebrate models, such as medaka fish, are frequently used as disease models to investigate how a disease affects organ function, which is usually caused by altered organ morphology. Thus, the described pipeline will allow obtaining vital information on disease etiology by studying disease phenotypes via changes in morphometric properties of organs.

4.2. 3D hybrid fibrous scaffolds for bone tissue engineering

Bone is a sophisticated organ, which protects internal organs from damage and supports the entire body. It consists of collagen fibers impregnated with hydroxyapatite (HA) such as calcium and phosphate, which increase hardness and durability of bone tissue [379].

Bone tissue is continuously renewed during life through the remodeling process. However, this process can be unstable and may fail in a case of various bone disorders like fractures and diseases. Despite many bone treatment methods available today, they have a range of disadvantages, including the potential to trigger an immune response and the risk of disease transmission [380].

Alternatively, the new principles of tissue engineering can be employed in bone treatment. One of the promising directions of tissue engineering is the development of special constructions, so-called scaffolds [381].

These constructions have a particular structure, which mimics the structure and properties of natural bone tissue. Bone tissue has a specific sort of orientation depending on a type of bone (*e.g.*, cortical or trabecular bone). The controlling of these properties in the scaffolds fabrication is a crucial task in material science because it helps to regulate cells response during the regeneration process. Also, specific orientation of the scaffold structure may influence the mechanical properties of the construction, which is essential in the regeneration process of the bone tissue as well.

In this study, fabrication of scaffolds was performed via electrospinning technique [382]. This method was chosen due to its versatility and simplicity of the technological process. For mimicking the structure of different bone types, the randomly-oriented and aligned fiber structure was formed by controlling the rotation speed of a collector [383].

For scaffolds fabrication, the use of composite materials is highly recommended due to their advantageous properties compared to one single material. In this work, the combination of polycaprolactone (PCL) and silicate-containing hydroxyapatite (SiHA) biomaterials was used. Previously it was shown that PCL polymer material could be successfully applied for scaffolds fabrication due to its high mechanical properties and a long period of degradation (2-3 years) [384], which is crucial in the regeneration process of bone tissue. However, due to its hydrophobic nature for enhancing bioactivity, different additives can be used [385, 386]. As bone tissue consists of HA, it can be added to the polymer solution for improvement of scaffold's properties. Si in the HA structure may improve and enhance the mineralization process of the bone and cause a positive effect in the rate of bone formation [387, 388].

The studying of scaffolds was performed with high-resolution SR-μCT, which advantages were previously described in Sec. 1.3.1. The analysis of morphological properties of fibrous scaffold structures presented in 3D datasets has been performed with the segmentation framework presented in Sec. 2.1, which specific implementation is illustrated in Fig. 4.11.
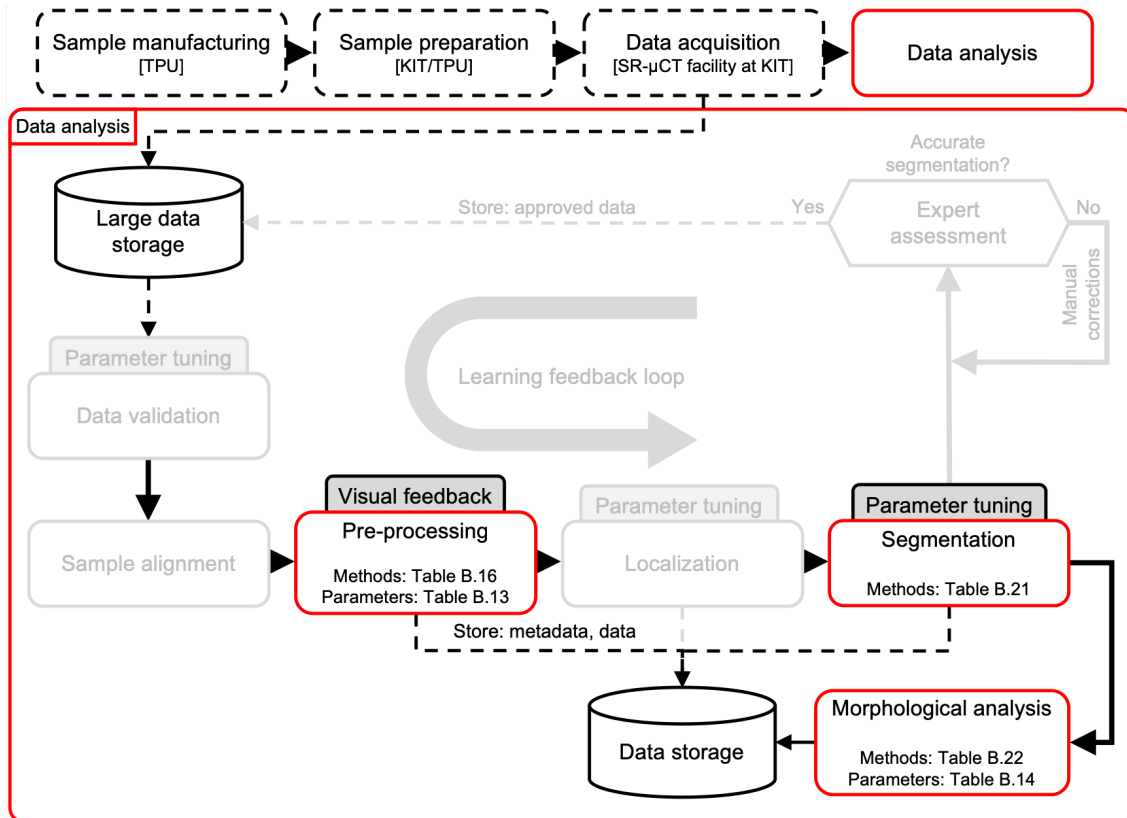
Figure 4.11.: The general diagram of the segmentation framework with the specific implementation and parameterization of each stage for µCT datasets of fibrous scaffolds. The data analysis stages are described with methods and parameters, while the data preparation stages are characterized by institutions and facilities where experiments were carried out (*e.g.*, KIT and Tomsk Polytechnic University (TPU)). The unused stages are marked in gray.

4.2.1. Methods

Scaffold fabrication

The detailed information about the electrospinning process and chemicals used for this experiment is presented elsewhere [382]. The fabrication process was performed at Tomsk Polytechnic University (TPU). The pure fibrous scaffolds with randomly oriented (rPCL) and well-aligned (wPCL) structures were fabricated using a PCL solution of 9% (w/v). For composite scaffolds with the same structure (rPCL-SiHA, wPCL-SiHA) the mixture of 9% (w/v) of PCL and 10 wt.% of precursor powder of SiHA was used. The electrospinning setup was adjusted at the following parameters: a solution feeding rate of 2 ml/h, a high voltage of 17 kV, and a distance between a needle tip and a collector of 7 cm. The rotation speed of the collector was fixed at 600 and 1000 rpm for fabrication randomly oriented and aligned structures, respectively. The duration of samples formation was set at 1 hour.

SR-µCT imaging

The high-resolution µCT was performed at the bending-magnet imaging station at the synchrotron light source facility at KIT [374]. The samples were scanned with the monochromatic beam of 12 keV energy, which was converted into visible light with $Lu_3Al_5O_{12}$ scintillator of 200 µm. The X-ray projections were detected by an sCMOS camera (2560×2160 pixels, 6.5×6.5 $µm^2$ pixel size) in conjunction with an optical light 3.6x microscope providing a spatial resolution of 1.8 µm and a visual field of view of 4.6×3.9 mm. The samples were rotated with a step of 0.24° and exposed for 1 sec for the beam, totally 1500 projections were acquired over 360° rotation per sample. The projections were flat-field corrected to reduce the noise of camera and imperfections of the scintillator. The tomographic reconstruction was performed using the FBP algorithm implemented in the UFO framework [376].

Data segmentation and analysis

The segmentation framework, presented in (Sec. 2.1) was utilized in data pre-processing, segmentation and further morphological analysis. The analysis of the reconstructed datasets consisted of quantification of the orientation and diameter distribution of the fiber structures, as well as a size distribution of inclusions, and the determination of the porosity parameter of each sample.

Due to the specificity of data analysis task, only a part of processing stages was used, namely, pre-processing, non-learning segmentation and morphological analysis. The stages comprising framework are decoupled from each other that allows them to be connected arbitrarily. In the morphological stage, only the quanfima module was utilized for fiber analysis.

Pre-processing The reconstructed datasets were passed through the denoising functionality of the pre-processing stage configured as Tab. B.14 to reduce statistical noise and provide homogeneity of structures for the further segmentation stage. The optimal denoising parameters: the smoothing factor $\gamma_{sm} = 4$, the patch size $W_{tp} = 5$, and the search window $W_{sw} = 21$ were found empirically from results produced by the denoising method over a set of various combination of parameters for representative data slices.

Segmentation The denoised datasets were then segmented to separate fibers and particles from a background. The histogram-based segmentation method has been empirically selected because there are no strict rules in algorithm selection, rather recommendations (Sec. 2.4). The density distributions of fibers and particles are completely different; therefore they require individual segmentation methods. The particles were segmented with threshold-based maximum entropy method, whereas fibers with the Otsu thresholding method. The algorithms were

applied to every slice of a dataset separately in a slice-wise fashion along the Z-axis. The produced segmented slices formed a dataset with physically separated particles and fibers.

Morphological analysis   The segmented particles and fibers were transferred to the next stage to perform the morphological analysis, where several aspects of fibers and particles were estimated, namely, fiber orientation and diameter, the number of particles and their volumes, and porosity of a dataset.

Before proceeding with the orientation analysis, the segmented fibers were skeletonized to extract the medial axis as described in (Sec. 3.1). Then, the ray-casting method run over the skeletonized segmentation at every point of the skeleton. The algorithm described in (Sec. 3.2.3) estimated orientation as a pair of azimuth and elevation angles in the neighborhood of a sphere of diameter $L = 32$ centered at every skeleton point, with steps of scanning angular ranges of $\theta_\Delta = 1°$ and $\varphi_\Delta = 1°$. When the orientation of fibers were estimated, the diameter could be calculated using the approach (Sec. 3.3.2).

The segmented particles were counted and estimated individually for volume and diameter with the connected-component analysis mentioned in (Sec. 3.3.2). The porosity for each dataset was calculated from the combined segmentations of fibers and particles as described in Sec. 3.3.2.

4.2.2. Characterization of the structure and morphology

The morphology is one of the critical factors influencing cells adhesion and proliferation [389]. The addition of SiHA to polymer solution may distort fiber diameter in scaffolds. This phenomenon was observed by Metwally *et al.*, who fabricated PCL scaffolds with embedded pure HA and calcium carbonate microparticles using electrospinning technique [390].

The reconstructed datasets were cropped for better representation to the ROIs of $400 \times 400 \times 230$ pixels, which correspond to $720 \times 720 \times 414$ µm and are presented in Fig. 4.12. The visualizations demonstrate that fabricated PCL and PCL-SiHA scaffolds possess the expected fiber morphology, which would correlate to the natural structure of bone tissue.
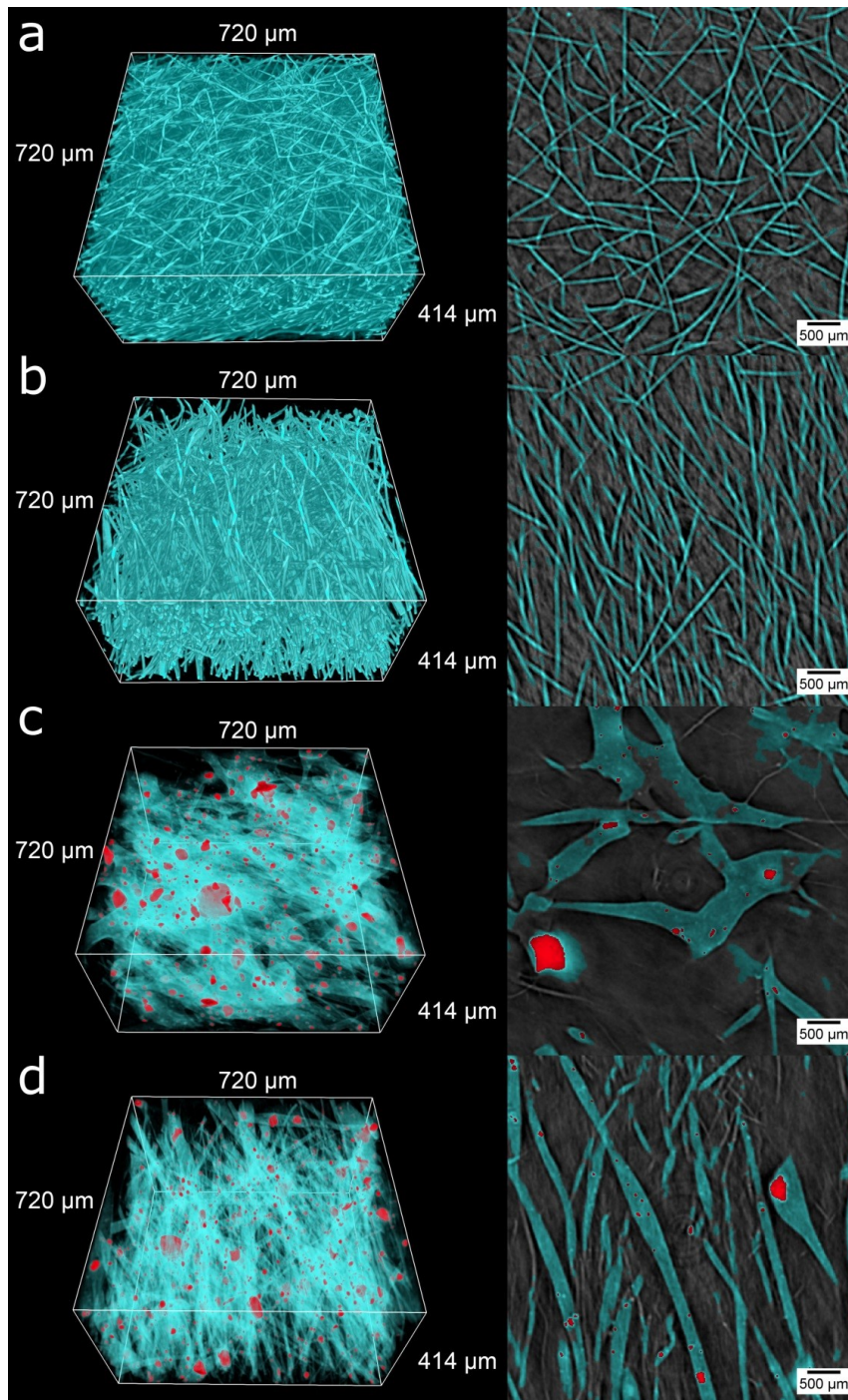
Figure 4.12.: The visualization of polymer PCL and hybrid PCL-SiHA scaffolds with SR-μCT. The 3D renderings and corresponding axial slices along the height of samples are presented in the left and right columns, respectively: (a) rPCL, (b) wPCL, (c) rPCL-SiHA and (d) wPCL-SiHA. The fibers are colored in cyan, while microparticles and their aggregates are in red.

In the studied scaffolds, there were detected changes in the fiber morphology for samples containing SiHA microparticles. A rough surface morphology with

massive polymer structures was observed in the randomly oriented (r) and well-aligned (w) samples, rPCL-SiHA and wPCL-SiHA, correspondingly. The SiHA microparticles intensively agglomerate over the entire volume. The most massive aggregates tended to embed within an even larger polymer structure. These drastic morphology changes may be related to change in electrostatic forces since Si inclusions are non-conductive and/or increase viscosity regarding pure PCL.

The azimuthal and elevational fiber orientation histograms are presented in Fig. 4.13. In the azimuthal direction (Fig. 4.13a), scaffolds with a well-aligned structure (wPCL) have a prevailing number of fibers oriented from 75° to 100°. While scaffolds with a randomly oriented structure (rPCL) have a predominant directionality ensured by a rotating collector, however, in wider angular ranges, the fibers of rPCL are mostly oriented in the range from 80° to 160°, and rPCL-SiHA in the ranges from 20° to 60° and from 120° to 180°. As can be seen, all samples in the elevational direction have similar orientation from 50° to 90°, which is dictated by a layer-by-layer fiber deposition during the electrospinning process (Fig. 4.13b).
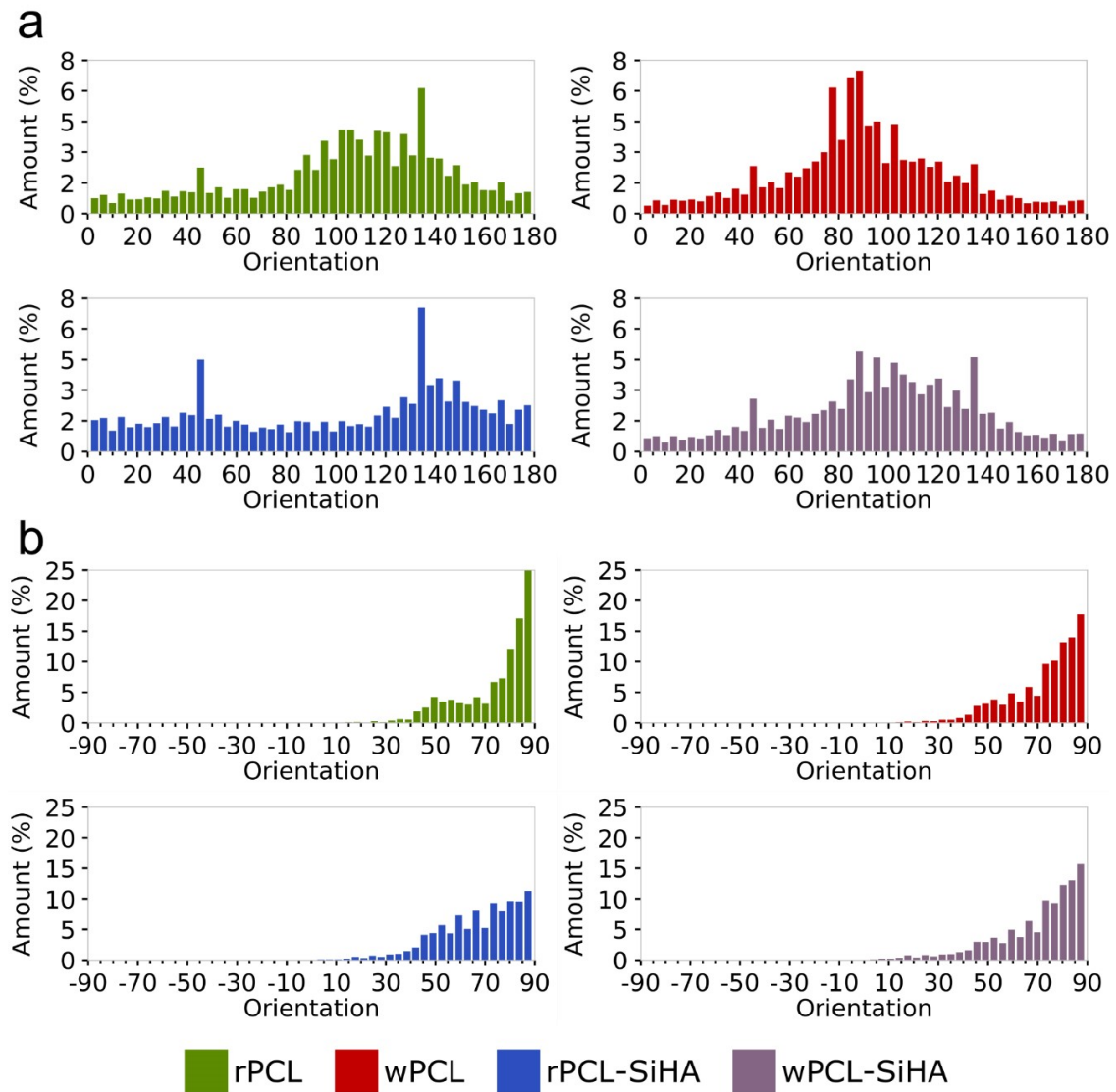
Figure 4.13.: The orientation histograms of polymer PCL and hybrid PCL-SiHA scaffolds in (a) azimuthal and (b) elevational directions.

The complete analysis of fiber orientation presented in Fig. 4.14 grants an immediate visual comparison of scaffolds with randomly oriented and well-aligned fiber orientations, where similar colors represent fibers aligned in a similar direction. Notably, fibers comprising wPCL change their dominant orientation depending on sample height, which is evidence of a difference in layer deposition during electrospinning. Moreover, the visualizations demonstrate that most fibers comprising scaffolds with an aligned structure lay in the same plane, while fibers of scaffolds with a randomly oriented structure are chaotically distributed in a volume.
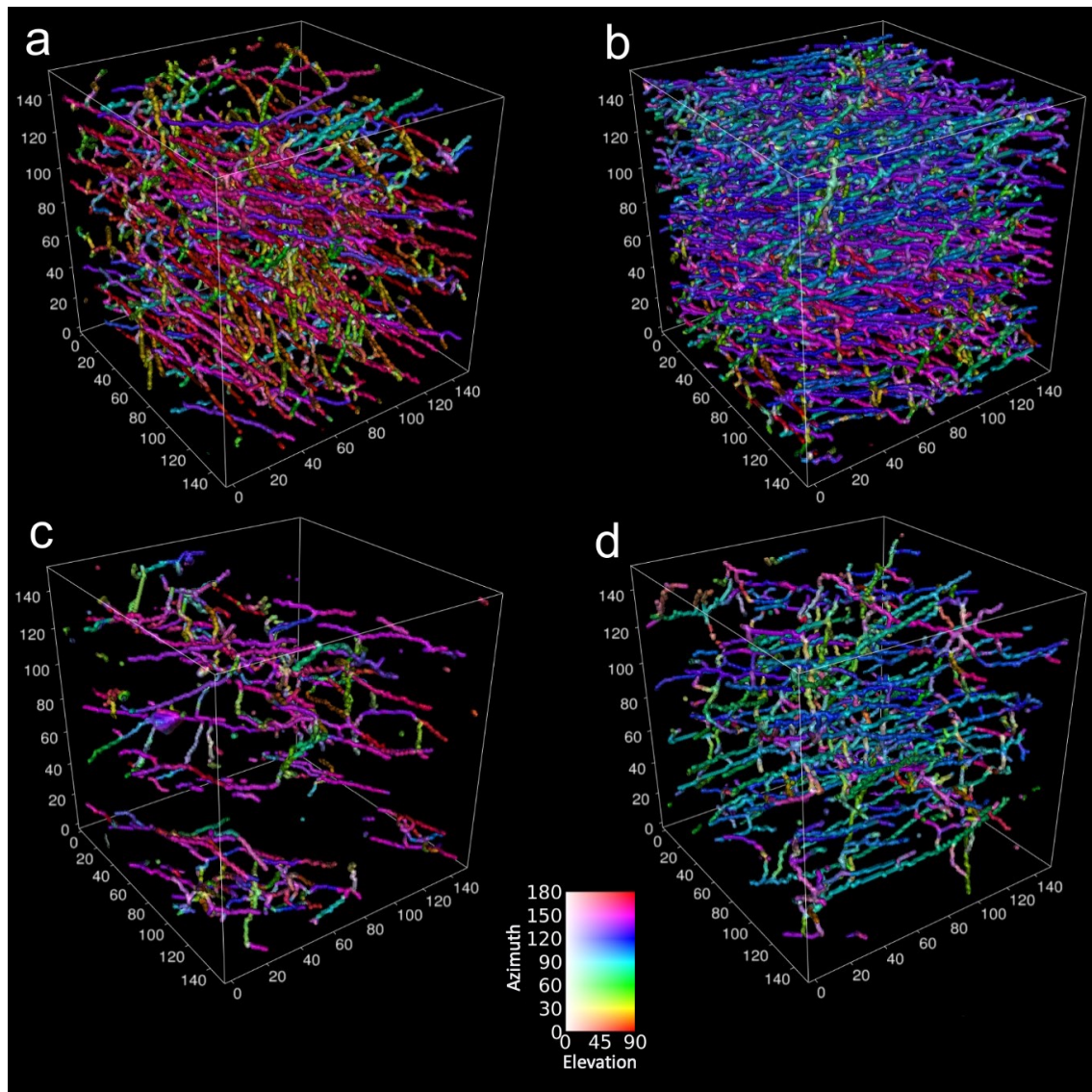
Figure 4.14.: The visualization of fiber orientation for scaffolds: (a) rPCL, (b) wPCL, (c) rPCL-SiHA and (d) wPCL-SiHA.

The results of fiber diameter analysis are presented in Fig. 4.15 showed that hybrid scaffolds have the thickest fibers in the range from 35 µm to 75 µm, while the largest number of fibers presented in the range from 10 µm to 25 µm. It was also found out that the fiber diameter of wPCL and wPCL-SiHA scaffolds decreased relatively to rPCL and rPCL-SiHA scaffolds. The pie charts illustrate that the fiber diameter of wPCL increased in the range from 1 µm to 5 µm for 6.7% and decreased from 10 µm to 35 µm for 2.7%. This study revealed that scaffolds with SiHA microparticles have a similar trend, where the fiber diameter of wPCL-SiHA increased in the range from 1 µm to 10 µm for 0.8% and from 10 µm to 25 µm for 14.4%, and decreased from 25 µm to 35 µm for 1.3%, from 35 µm to 45 µm for 5.1%, and from 45 µm to 75 µm for 8.7%, relative to the rPCL-SiHA scaffold. This behavior of fiber diameter can be explained by excessive stretching of

a solution jet before to be deposited onto collector rotating at high speed.
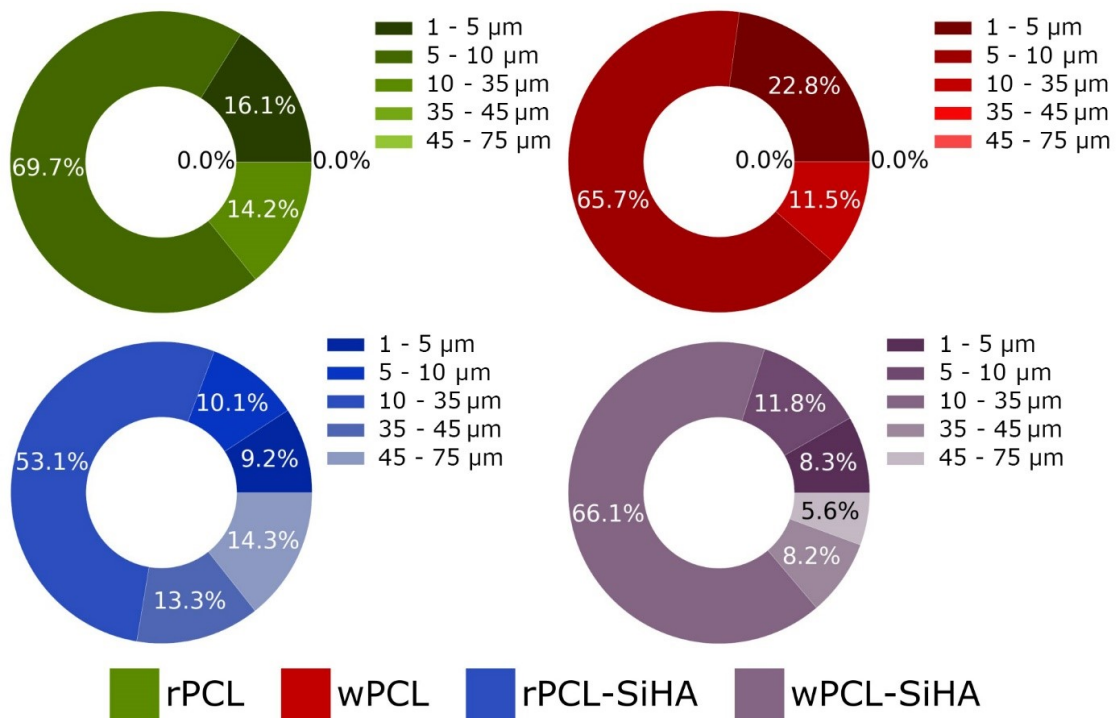
Figure 4.15.: The visualization of fiber diameter analysis for SR-µCT datasets of polymer PCL and hybrid PCL-SiHA scaffolds.

The size distributions of SiHA microparticles and its aggregates in rPCL-SiHA and wPCL-SiHA scaffolds are presented in Fig. 4.16(a,b). The analysis has shown that the most frequently detected microparticle aggregates were in the range from 5.83 µm^3 to 1000 µm^3, which corresponds to an approximate diameter from 2.23 µm to 12.4 µm, respectively, assuming that aggregates are modeled with spheres. The larger aggregates were detected in the range from 1000 µm^3 to 1000000 µm^3, which corresponds to an approximate diameter from 12.4 µm to 124 µm, but their total number was not higher than 13.6% for rPCL-SiHA and 13.8% for wPCL-SiHA. However, microparticles of volume less than 5.83 µm^3 could not be resolved with this setup due to limits of spatial resolution of 1.8 µm. In the case of both hybrid types of scaffolds, a similar number of microparticles and their aggregates were observed, and no statistical significance was recognized (Fig. 4.16c).

4. Applications
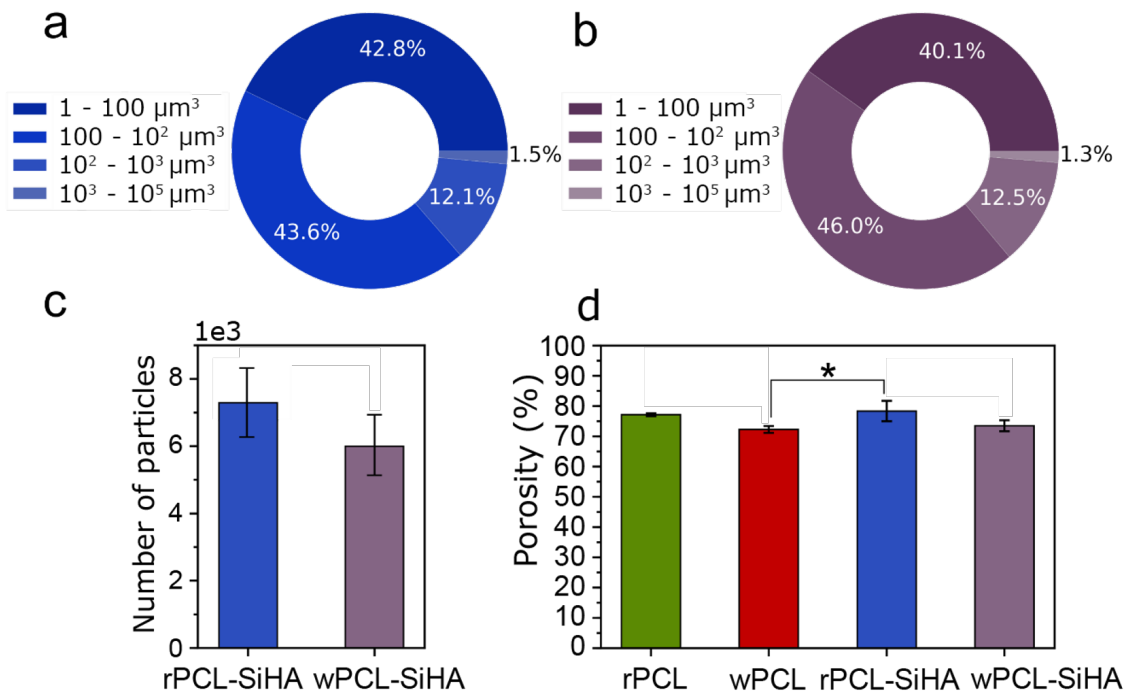
Figure 4.16.: The results of structural analysis of the scaffolds from SR-µCT datasets. The pie charts show percentage of the total number of microparticles and their aggregates in each size range for (a) rPCL-SiHA and (b) wPCL-SiHA; (c) histogram displays the total number of inclusions presented in hybrid scaffolds and (d) histogram shows porosity of polymer and composites (*p $\leq$ 0.05 and N.S. with p > 0.05).

The composite scaffolds have shown slightly increased porosity of 78.36$\pm$3.35% and 73.50$\pm$1.87% for rPCL-SiHA and wPCL-SiHA, respectively, whereas, rPCL and wPCL pure scaffolds have demonstrated porosity of 77.14$\pm$0.49% and 72.27$\pm$1.14%, correspondingly (Fig. 4.16d). As can be seen from the results of the analysis, SiHA inclusions influence change in porosity of scaffold structure. These facts can help to optimize a scaffold design, which may enhance attachment and proliferation of cells into scaffolds [391].

Discussion

The data analysis workflow for datasets produced from SR-µCT experiments over samples of simple structure, *i.e.*, composed of a few materials was created from the processing stages provided by the proposed segmentation framework in Sec. 2.1. The processing stages are decoupled from each other, therefore, they can be connected arbitrarily to provide the analysis workflow of the required structure. Moreover, the functionality within each stage is also decoupled, that allows providing higher flexibility, *i.e.*, only a part of operations within a stage can be executed if necessary.

The pre-processing was required only to reduce statistical noise with the denoising functionality while skipping all other preparation steps for further segmentation with CNNs. Since the datasets were composed of several homogenous materials, which resulted in multi-modal intensity distributions, they did not require any sophisticated CNN-based segmentation approaches. Therefore, intensity histograms of the datasets could be analyzed with histogram-based segmentation methods to find correct thresholding values to separate materials from each other.

The morphological analysis stage was as well only partially utilized because only analysis of fibers and particles was required. The fibers were subjected to the specific kind of analysis, while particles were analyzed with the volumetric analysis functionality.

4.2.3. Conclusion

In this section, the characterization of PCL scaffolds with randomly oriented and well-aligned fiber structures aimed at the regeneration of bone tissue was presented. The embedding of SiHA microparticles and their aggregates into a structure of scaffolds allowed to improve cellular penetration and bone in-growth. The proposed segmentation framework was utilized to analyze SR-μCT datasets of scaffolds for the fiber diameter and orientation, porosity, and the distribution of inclusions size. The results demonstrated that more often SiHA microparticles were detected in the range from 2.23 μm to 12.4 μm, which influences the fiber morphology and has little impact on the orientation of fibers. Thus, the presence of inclusions changes the morphological properties of the material, while it does not influence on the orientation of fibers.

5. Conclusions

The development of µCT imaging techniques and instrumentation, digital detectors, and smart control systems has led to the emergence of high-performance imaging setups capable of producing a tremendous amount of data per scanning session. The application of such imaging setups in high-throughput experiments involving a large number of complex heterogeneous specimens, as in a case of studies for phenotype-genotype association or drug action on morphometry of living organisms, produce dozens of terabytes of data, which should be processed and analyzed quickly to take the next step in the planning of research. Such amount of data are impossible to handle manually since even the domain experts can easily spend weeks on a manual delineation of target structures and subsequent analysis. While, since the experts are not machines, they tend to be prone to a human factor which incorporates mistakes into a segmentation process. Moreover, manual delineation process is subjective, because different experts may see boundaries of target structures differently. The available automated analysis approaches are not applicable for such complex heterogeneous datasets due to their simplicity, inability to exploit the hierarchical nature of specimens, lack of flexibility and low performance. The primary aim of this thesis was to develop a new concept for the automated analysis of complex specimens to provide a fast and reliable workflow for various high-throughput X-ray experiments. This goal was accomplished by developing a new concept for hierarchical data analysis, improving reliable state-of-the-art approaches from different application fields aimed at registration, localization, and segmentation to fit specificity of such data, and developing a new method for more precise morphological analysis of fibrous structures.

The major contributions presented in this thesis can be formulated as follows:

1. A new concept for hierarchical data analysis of complex heterogeneous datasets of similar nature acquired in high-throughput experiments employing laboratory or synchrotron-based µCT imaging setups (Chap. 2).

2. A new automated analysis workflow tailored for efficient processing of complex heterogeneous µCT datasets of hierarchical nature (Sec. 2.1). The proposed workflow is based on improved state-of-the-art approaches for validation, pre-processing, registration, localization, and segmentation, where any stage involving a training phase can be automatically fine-tuned using a grid search approach over the specified parameter space (Sec. 2.1.3).

3. A new, rapid, highly parallelizable 3D orientation method for analysis of fibrous structures in large µCT datasets (Sec. 3.2.3). The proposed method provides higher accuracy than the state-of-the-art tensor-based method and outperforms it by a factor of twenty using the GPU implementation, which increases with the size of the dataset (Sec. 3.4).

4. A new algorithm for generation of a synthetic dataset composed of non-/overlapping straight fiber-like structures with such properties as spatial location, length, orientation, radius, a minimum distance to neighbor structures sampled from the specified ranges (Sec. 3.2.4). The proposed algorithm allows generating ground-truth datasets with fibers having specific properties to validate methods for 3D orientation analysis.

5. The thorough evaluation of the developed workflow on the phantom datasets simulating results of µCT experiments (Sec. 2.9). Every stage of the workflow utilizing neural networks was assessed individually for various network architectures, data augmentation strategies and noise robustness to determine the best-suitable parameters.

6. The comparative analysis of the developed 3D orientation analysis method to the tensor-based approach in terms of accuracy and throughput (Sec. 3.4) on the synthetic dataset generated with the algorithm proposed in (Sec. 3.2.4). The synthetic dataset was comprised of tightly packed and non-overlapped fibers of highly varying parameters. The methods were evaluated on the generated datasets to determine the optimal parameters and assess robustness to noise. The throughput of methods was estimated for CPU and GPU implementations on a dataset of varying size and a fixed configuration of fibers.

7. The implementation of all stages of the developed workflow as independent modules to allow a user to combine them accordingly to the task and data on hands (Sec. 2.8). The independence of modules allows creating a workflow using any workflow management library [392]. The modules were developed using the Python language to allow execution on any platforms including Windows, Linux, and macOS and integration with Jupyter notebooks. The dependencies on the third-party libraries were reduced to ease the deployment process.

8. The implementation of Quanfima package aimed at a complete 2D and 3D morphological analysis of fibrous structures (Sec. 3.3). The proposed 3D orientation analysis method (Sec. 3.2.3) was implemented as a part of the Morphology module of this package and available to be executed on both CPU and GPU. The package was implemented with the Python language to provide immediate integration into other workflows and be used with interactive Jupyter notebooks.

9. The successful application of the newly developed and implemented automated analysis workflow to analyze the high-resolution datasets of medaka fish acquired with the synchrotron-based µCT imaging setup at KIT in the

frame of the phenotyping experiment (Sec. 4.1). The brain, liver, head nephrons and heart from 20 medaka fishes were segmented and analyzed automatically with the proposed workflow, which paves the way for unraveling correlations between the genetics and the morphological characteristics of various organs.

10. A complete morphological analysis including orientation, diameter, and density of the polycaprolactone 3D scaffolds using the implemented Quanfima package (Sec. 4.2). The developed 3D orientation estimation method was employed for orientation analysis. The quantitative results of the analysis allow guiding the fabrication process to produce scaffolds with desirable mechanical properties.

The X-ray imaging techniques are continuously developing to provide faster acquisition and higher quality of the acquired data. However, along with 3D X-ray imaging which captures a static specimen, 4D (3D+t) X-ray imaging is also actively developing to digitalize a specimen in motion to discover dynamic processes behind its complex structure. A biological screw joint of *Trigonopterus oblongus* [393] is a remarkable example of how 4D X-ray imaging helps to reveal hidden mechanisms. Such imaging techniques require analysis approaches capable of providing sufficient quality and performance. The proposed automated analysis workflow should be revised to meet new requirements imposed by 4D X-ray imaging. All stages should take into account the temporal dimension since the specimen is not static anymore. The validation stage should be able to track abnormalities along all temporal states of the specimen and exclude datasets which are outliers with respect to the initial state. The alignment stage should perform registration of the initial state of the specimen to the reference atlas, and then other datasets from the corresponding temporal states should be registered sequentially to compensate possible anatomical variations. The localization stage should adjust 3D bounding boxes through all temporal states to take into account the specimen movements. The segmentation stage should use all temporal information to improve the prediction of the structures by propagating the segmentation probabilities to the next state. These improvements will allow for tracking how the morphological properties of structures change in time that gives more in-depth insight into the internal dynamic processes of the specific specimen.

A new experiment aimed at studying species which were not previously scanned usually suffers from a lack of data for training localization and segmentation neural networks at the respective stages. These networks must be trained at representative examples to be able to recognize similar patterns in newly scanned datasets of similar specimens. In this case, several datasets should be manually segmented by the domain experts to provide reliable ground-truths for training. Data augmentation allows increasing amount and variety of training data by applying random geometrical transformations to manually segmented datasets and corresponding labels. However, geometrical transformations cannot provide substantial deformation of the target data. Therefore more severe variations could be introduced by elastic deformations, which are widely applicable in histology or

cells analysis, where the shape is simple, and most shape variations are plausible. Whereas, in a case of model organisms like mouse, Zebrafish or medaka fish, the outer shape and shapes of internal organs are restricted and dependent on each other. Hence, random elastic deformations are not straightforwardly applicable, because they do not take into account how each deformation of a single organ impacts on the whole system, since all organs are interconnected. Thus, deformation models providing plausible deformations of the entire system of structures are required for more realistic data augmentation. To address this issue, statistical shape models of every organ and the outer shape can be created and assembled into a single system. Thus, data augmentation of manually segmented datasets can be performed using this system of deformable models, which guides plausible and complex deformations of the whole model organism.

The performance of the proposed method for orientation analysis of large datasets could be further improved by introducing a hierarchical analysis via a coarse-to-fine strategy. The scanning angular ranges can be shrunk depending on the current scale level. Thus, at the most coarse level, the angular ranges are broad and thoroughly scanned to determine a set of the closest candidates corresponding for the correct orientation. Then, the determined angles accompanied by the specified confidence intervals for a minimum and maximum angles are passed to the finer level, where the algorithm scans only the specified angular ranges. The process of scanning and determination of new scanning ranges repeats until it reaches the finest level, where the scanning ranges are significantly shrunk and require much less time to find the final orientation.

Thus, improvements in X-ray imaging techniques allow revealing so far hidden internal structure and dynamic processes. However, they produce data, whose amount gradually increases year by year, and without a proper interpretation from the domain experts, it is just a set of bytes. Hence, the produced data should be processed and analyzed promptly with the methods which can completely utilize the potential of imaging techniques, and provide the experts with valuable quantitative information.

A. Nomenclature and Symbols

In this section, the notation including the basic concepts and symbols used throughout the thesis is summarized. The nomenclature is inspired by the formulation given in [394].

- Lower case letters are used for scalars, parameters, indexing variables and functions (*e.g.,* γ, μ, σ are common names for parameters, $f(\cdot)$ is common name for functions or functionals, i, j, k, l are common names for indices)

- Upper case letters are used for number of entities (*e.g.,* N_c, N_{ax} for the number of classes and the number of dimensions, respectively)

- Upper case bold face letters represent matrices (*e.g.,* $\boldsymbol{S}, \boldsymbol{I_m}$)

- Lower case bold face letters represent vectors (*e.g.,* $\boldsymbol{x}, \boldsymbol{y}$ for column vectors and $\boldsymbol{x}^\top, \boldsymbol{y}^\top$ for row vectors)

- Accessing elements of data structures is denoted by square brackets (*e.g.,* $\boldsymbol{x}[1]$ accesses the first element of vector $\boldsymbol{x}$, $\boldsymbol{I}[1, 2]$ accesses the matrix element in the first row and second column)

- Elements topped with a hat refer to estimates of a certain quantity (*e.g.,* $\hat{x}$ represents an estimate of the scalar x, $\hat{\mathbf{x}}$ represents an estimate of vector x)

- Optimal solutions are indicated by an asterisk (*e.g.,* x^* would denote the optimal solution to $\operatorname{argmin}_{a,b} f(a, b)$)

A. Nomenclature and Symbols

Symbol	Description
$\lvert x \rvert$	Absolute value of x
$*$	Convolution operator. Discrete convolution is used for images.
ANN	**A**rtificial **N**eural **N**etwork
Al	Aluminium
$\mathcal{A}$	A surface area measure
$\mathrm{argmin}(\cdot), \mathrm{argmax}(\cdot)$	The minimization and maximization functions.
$\frac{\partial f}{\partial x}$	Partial derivative of function f with respect to x
2D	Two dimensional
3D	Three dimensional
3D+t	Three spatial dimensions and temporal dimension
[...]	An array of items
{ ... }	An unordered set of items
BRIEF	**B**inary **R**obust **I**ndependent **E**lementary Features
BM	**B**ending **m**agnet
B	3D bounding box
CCD	**C**harge-**c**oupled **D**evices
sCMOS	**S**cientific **C**omplementary **M**etal **O**xide **S**emiconductor-based Detectors
CNR	**C**ontrast-to-**N**oise **R**atio
CNN	**C**onvolutional **N**eural **N**etwork
CAD	**C**omputer-**A**ided **D**esign
CC	**C**ross-**C**orrelation
CPU	**C**entral **P**rocessing **U**nit
CT	**C**omputed **T**omography
$\mathcal{C}$	The registration cost function
C	Integer class label

$\boldsymbol{C}$	Encoded vector of classes
DSC	**Dice Score**
d	Network depth level
$\Delta_x, \Delta_y, \Delta_z$	Pixel size along x, y, z axes
$\exp(\cdot), e^x$	Exponential function
$f_{\mathrm{ray}}(\cdot)$	Ray-casting function
$f_{ring}(\cdot)$	Ring artifact filtering function
$f_{median}(\cdot)$	2D median filtering function
$f_{norm}(\cdot)$	Intensity normalization function
$f_{sparse}(\cdot)$	Function generating a sparse representation
$f_{scale}(\cdot)$	Scaling function to the specified size
$f_{ohe}(\cdot)$	One-hot encoding algorithm
$f_{loss}(\cdot)$	Optimization loss function
$f_{th}(\cdot)$	Thresholding function
$f_{nlm}(\cdot)$	Non-local means filtering function
$f_{sobel}(\cdot)$	Sobel operator function
$f_{gth}(\cdot)$	Global thresholding function
$f_{morph}(\cdot)$	Sequence of morphological operations
$f_{cca}(\cdot)$	Connected-components analysis function
$f_{snorm}(\cdot)$	Spatial normalization function
$f_{pyr}(\cdot)$	Function creating a multi-scale pyramid
$f_{mhe}(\cdot)$	Multi-hot encoding algorithm
$f_{ctx}(\cdot)$	Function creating a 3D spatial context
$f_{savg}(\cdot)$	Function averaging size of datasets in a population
$f_{snear}(\cdot)$	Function finding of the nearest divisible by two size
$f_{bscale}(\cdot)$	Function scaling coordinates of a bounding box by the specified scale factor
$f_{pext}(\cdot)$	Function extracting 3D patches from a dataset
$f_{pcmb}(\cdot)$	Function reconstructing a dataset from 3D patches

A. Nomenclature and Symbols

$f_{fuse}(\cdot)$	Function creating a 3D probability map by averaging probabilities at every voxel
FBP	**F**iltered **B**ack **P**rojection
F_{ij}^{k}	The k-th feature map whose a receptive field centered at the location (i, j)
$\boldsymbol{G}(\boldsymbol{x}, \sigma)$	Gaussian kernel with standard deviation σ
GPU	**G**raphics **P**rocessing **U**nit
H	Shannon entropy
$\boldsymbol{H}$	Multi-scale pyramid
i, j, k, l, m, n	Index variables defined by the context
$\boldsymbol{I_m}$	Identity matrix
$\boldsymbol{I}$	General dataset variable
$\boldsymbol{L}$	General label variable
$\boldsymbol{I}'$	Changed dataset, determined by a context
$\boldsymbol{I}^{ax}$	Slices of a reconstructed dataset along axis ax
$\boldsymbol{I}_{sino}$	Sinogram image
$\boldsymbol{I}_{reco}$	Reconstructed slices forming a X-ray dataset
$\boldsymbol{I}_{norm}$	Intensity-wise normalized dataset
$\boldsymbol{I}_{scaled}$	Scaled dataset
$\boldsymbol{I}_{sparse}$	Sparse representation of a dataset
$\boldsymbol{I}_{val}$	Validated dataset
$\boldsymbol{I}_{M}, \boldsymbol{Y}$	Moving dataset
$\boldsymbol{I}_{F}, \boldsymbol{X}$	Fixed / reference dataset
$\boldsymbol{I}_{edge}$	Dataset with enhanced edges
$\boldsymbol{I}_{th}$	Thresholded dataset
$\boldsymbol{I}^{B}$	Data contained by a bounding box B
$\boldsymbol{I}_{sn}$	Spatially normalized dataset
$\boldsymbol{I}_{cts}^{ax}$	Dataset composed of 3D slices along the orthogonal axis ax
$\boldsymbol{I}^{P_k}$	The k-th 3D patch of size S_{pth}

ITK	**I**nsight **T**ool**k**it	
ID	**I**nsertion **D**evice	
ICP	**I**terative **C**losest **P**oint	
$\min(\cdot), \max(\cdot)$	Minimum and maximum functions	
MRF	**M**arkov **r**andom **f**ield	
MI	**M**utual **I**nformation	
MRI	**M**agnetic **R**esonance **I**maging	
N	General number of angles, objects or dimensions specified by the context	
NMI	**N**ormalized **M**utual **I**nformation	
N_d	Number of dimensions / Network depth	
N_{ax}	Number of dimensions	
N_{wl}	Number of wavelet decomposition levels	
N_f	General number of features	
N_m	Number of materials making up a sample	
N_x, N_y, N_z	Number of pixels in x, y, z direction	
N_ϕ	Number of elevation angles	
N_θ	Number of azimuth angles	
N_{mp}	Minimal number of projections	
N_{px}	Number of pixels in a row of each projection	
N_{ws}	Size of a local 3D window	
N_{qss}	Number of equally-spaced slices along the selected orthogonal axis	
N_{bf}	Number of filters at the first convolution layer	
N_{fc}	Number of neurons at the fully-connected layer	
N_{dp}	Amount of dropout	
N_s	Number of observations	
N_c	Number of classes or categories	
N_{mi}	Number of iterations of morphological operations	

A. Nomenclature and Symbols

N_L	Number of layers for a multi-scale pyramid
N_{ns}	Number of neighbor slices
N_{ss}	Size of slice step to form a 3D spatial context
N_{caa}	Number of extracted connected components
N_{pths}	Number of extracted 3D patches from a dataset
L_{ray}	Length of X-ray path transmitting through a sample
L	Diameter of a restricting sphere for a casted ray
$\mathcal{N}(\mu, \sigma)$	Normally distributed random variable with mean μ and standard deviation σ
PCL	**P**oly**c**apro**l**actone
PTA	**P**hospho**t**ungstic **A**cid
P_{low}, P_{high}	Percentiles defining an intensity range for conversion
$\boldsymbol{P}^{ax}$	Prediction vector along the orthogonal axis ax
$\boldsymbol{P}$	3D probability map
ROI	**R**egion **o**f **I**nterest
RF	**R**andom **f**orest
RAM	**R**andom **A**ccess **M**emory
ReLU	**R**e**c**tified **L**inear **U**nit
R	Connected component represented as a range of indices
$\boldsymbol{R}$	Vector of connected components
ρ	Length of a fiber
$\mathfrak{R}_{ij}$	Array of indices belonging to some local region centered at the location (i, j)
$\boldsymbol{S}_r$	Euclidean sphere structuring element with radius r
SDK	**S**oftware **D**evelopment **K**it
$\mathcal{S}$	Sphericity measure
S	General variable for size
S_{mrf}	Minimal size of the receptive field at the last convolution layer
S_{val}	Size of a dataset at the validation stage

S_{se}	Structural element of morphological operations
S_{norm}	Average size of datasets in a population
S_{seg}	Size of a dataset at the segmentation stage
S_{pth}	Size of 3D patches extracted from a dataset
$\boldsymbol{S}$	Second-order structure tensor matrix
SNR	**S**ignal-to-**N**oise **R**atio
SIFT	**S**cale-**I**nvariant **F**eature **T**ransform
SURF	**S**peeded-**U**p **R**obust **F**eatures
SSD	**s**um of **s**quared **d**ifferences
$\text{sgn}(\cdot)$	Signum function that returns the sign of a real number
SR-µCT imaging	**S**ynch**r**otron-µCT imaging
SiHA	**Si**licate-containing **H**ydroxy**a**patite
VTK	**V**isualization **T**ool**k**it
$\mathcal{V}$	Volume measure
$\boldsymbol{T_{\mu}}$	Geometric transformation parametrized by $\boldsymbol{\mu}$
$\boldsymbol{W}_k$	The k-th feature map related to the specific filter
$W \times H \times D$	The width, height and depth of a dataset
W_{med}	Window size of the median filter
W_{sw}	Size of search window of the non-local means algorithm
W_{tp}	Patch size of the non-local means algorithm
$\mathcal{WD}(\cdot)$	Wall distance measure
γ_{sm}	Smoothing parameter of the non-local means algorithm
γ_{th}	Thresholding value
γ_{ds}	Pyramid downscaling factor
γ_{ovlp}	3D patch overlap
γ_{seg}	Thresholding value to generate a binary segmentation
γ_{rad}	Maximum length of ray propagation
θ	Azimuth angle of a fiber
θ_{Δ}	Step of azimuth angle

A. Nomenclature and Symbols

φ_Δ	Step of elevation angle
$\lambda_1, \lambda_2, ..., \lambda_{N_d}$	Eigenvalues sorted in ascending order of a $N_d \times N_d$ matrix defined in the context
$\boldsymbol{\lambda}$	N_d-dimensional vector of eigenvalues sorted in ascending order
μ	Arithmetic mean value
π	Mathematical constant (approximately 3.14159)
φ	Elevation angle of a fiber
σ	Standard deviation
σ_{agn}	Zero-mean additive Gaussian noise standard deviation
σ_{smooth}	The standard deviation of a Gaussian smoothing kernel
σ_{dmp}	The standard deviation of a Gaussian dumping function
σ_{tsr}	The standard deviation of a Gaussian smoothing kernel for a second-order structure tensor matrix
σ_{pyr}	The standard deviation of a Gaussian smoothing kernel of a multi-scale pyramid
σ_{pvs}	The standard deviation of a Gaussian smoothing kernel for a prediction vector

B. Tables

The chapter presents the tables related to the performance evaluation (Sec. 2.9) and applications (Chap. 4).

B.1. Performance evaluation

B.1.1. The validation stage

Parameter	Value
Median filter size (W_{med})	3 pixels
Scaled dataset size (S_{val})	$128 \times 128 \times 128$ voxels
Step between equally-spaced slices (N_{qss})	10%

Table B.1.: The pre-processing parameters of the validation stage for the performance analysis.

Parameter	Value
Number of base filters (N_{bf})	4, 8, 16
Minimal size of the receptive field (N_{mrf})	4, 8, 16
Dropout ratio (N_{dp})	0.5
Number of neurons (N_{fc})	1000
Number of epochs	250
Batch size	8
Weights initialization	Glorot initialization
Optimization method	Adam
Learning rate	0.0001
Learning rate reducing factor	2
Number of patience epochs	10

Table B.2.: The parameters of validation networks to determine the optimal architecture.

B.1.2. The alignment stage

Parameter	Value
Number of levels	4
Scale factors	8, 4, 2, 1
Smoothing factors	3, 2, 1, 0
Optimization method	Gradient descent
Cost function	Mutual information
Stop rule value (ϵ)	0.0001
Number of iterations	1000, 1000, 1000, 500

Table B.3.: The parameters of the registration method for the alignment stage.

B.1.3. The pre-processing stage

Parameter	Value
Smoothing factor (γ_{sm})	7
Patch size (S_{val})	3 pixels
Search window size (N_{qss})	21 pixels
Structural element size (S_{se})	5 pixels
Spatially normalized dataset size (S_{norm})	$512 \times 512 \times 512$ voxels
Number of iterations of morphological filters (N_{mi})	3
Number of pyramid levels (N_L)	4
Pyramid downscale factor (γ_{ds})	2.0
Pyramid smoothing factor (σ_{pyr})	1.0

Table B.4.: The parameters of the pre-processing stage during the performance analysis.

B.1.4. The localization stage

Parameter	Value
Number of base filters (N_{bf})	4, 8, 16
Minimal size of the receptive field (N_{mrf})	4, 8, 16
Dropout ratio (N_{dp})	0.5
Number of neurons (N_{fc})	1000
Number of epochs	250
Batch size	8
Weights initialization	Glorot initialization
Optimization method	Adam
Learning rate	0.0001
Learning rate reducing factor	2
Number of patience epochs	10
Number of neighbor slices (N_{ns})	3, 5
Step between neighbor slices (N_{ns})	1, 3

Table B.5.: The parameters of localization networks to determine the optimal architecture.

Parameter	Value
Translation	(-0.25, 0.25)
Rotation	(-15°, 15°)
Scaling	(0.75, 1.25)

Table B.6.: The parameters of the affine augmentation strategy of convolution networks during the performance analysis.

B.1.5. The segmentation stage

Parameter	Value
Number of base filters (N_{bf})	4, 8, 16
Minimal size of the receptive field (N_{mrf})	4, 8, 16
Number of epochs	250
Batch size	8
Weights initialization	Glorot initialization
Optimization method	Adam
Learning rate	0.0001
Learning rate reducing factor	2
Number of patience epochs	10

Table B.7.: The parameters of segmentation networks to determine the optimal architecture.

B.2. Quantitative morphometric analysis of adult teleosts

B.2.1. The pre-processing stage

Parameter	Value
Smoothing factor (γ_{sm})	7
Patch size (S_{val})	3 pixels
Search window size (N_{qss})	21 pixels
Structural element size (S_{se})	5 pixels
Spatially normalized dataset size (S_{norm})	1152×1024×6656 voxels
Number of iterations of morphological filters (N_{mi})	3
Number of pyramid levels (N_L)	6
Pyramid downscale factor (γ_{ds})	2.0
Pyramid smoothing factor (σ_{pyr})	1.0

Table B.8.: The parameters of the pre-processing stage for the application on medaka fish.

B.2.2. The localization stage

Parameter	Value
Number of base filters (N_{bf})	16
Minimal size of the receptive field (N_{mrf})	4, 8, 16
Dropout ratio (N_{dp})	0.5
Number of neurons (N_{fc})	1000
Number of epochs	250
Batch size	8
Weights initialization	Glorot initialization
Optimization method	Adam
Learning rate	0.0001
Learning rate reducing factor	2
Number of patience epochs	10
Number classes (N_c)	3
Number of neighbor slices (N_{ns})	5
Step between neighbor slices (N_{ns})	1

Table B.9.: The parameters of localization networks to determine the optimal architecture for the application on medaka fish.

Parameter	Value
Translation	(-0.15, 0.15)
Rotation	(-25°, 25°)
Scaling	(0.75, 1.25)

Table B.10.: The parameters of the affine augmentation strategy of convolution networks during the application on medaka fish.

Network name	Average wall-distance (µm)
CNN[8MRF 16BF 3NS 1SS]	**10.89±12.42**
CNN[4MRF 16BF 3NS 1SS]	16.11±17.85
CNN[2MRF 16BF 3NS 1SS]	19.42±21.57

Table B.11.: The accuracy of localization for different networks configurations.

B.2.3. The segmentation stage

Parameter	Value
Number of base filters (N_{bf})	16
Minimal size of the receptive field (N_{mrf})	4, 8, 16
Number of epochs	250
Batch size	8
Weights initialization	Glorot initialization
Optimization method	Adam
Learning rate	0.0001
Learning rate reducing factor	2
Number of patience epochs	10
Number classes (N_c)	1

Table B.12.: The parameters of segmentation networks to determine the optimal architecture for the application on medaka fish.

B.3. 3D biodegradable scaffolds of polycaprolactone for bone tissue engineering

B.3.1. The pre-processing stage

Parameter	Value
Smoothing factor (γ_{sm})	4
Patch size (S_{val})	5 pixels
Search window size (N_{qss})	21 pixels

Table B.13.: The parameters of the pre-processing stage for the application on the PCL scaffolds.

B.3.2. The morphological analysis stage

Parameter	Value
Neighborhood sphere diameter (L)	32
Step size of scanning angular ranges ($\theta_\Delta, \varphi_\Delta$)	(1°, 1°)

Table B.14.: The parameters of the morphological analysis stage for the application on the PCL scaffolds.

B.4. Implementation description of the segmentation framework

Method description	Function name
Ring artifact filtering (Eq. (2.1))	`ring_filter`
Data denoising with a median filter (Eq. (2.2))	`median_filter`
Intensity normalization (Eq. (2.3))	`normalize`
Data rescaling (Eq. (2.4))	`rescale`
Input data preparation (sparse representation) (Eq. (2.5) and (2.6))	`validation_data_prep`
Model training process (Eq. (2.7))	`DataValidator.train`
Model prediction process (Eq. (2.8) and (2.9))	`DataValidator.validate`

Table B.15.: The correspondence between mathematical equations and implemented functions for the data validation stage.

Method description	Function name
Align the target dataset to the reference one (Eq. (2.10) to (2.12), (2.14) and (2.15))	`DataAligner.align`
Apply geometric transformation to the target dataset (Eq. (2.13))	`DataAligner.apply_transform`

Table B.16.: The correspondence between mathematical equations and implemented functions for the sample alignment stage.

Method description	Function name
Non-local means filtering (Eq. (2.16))	`nlm_filter`
Sobel filtering (Eq. (2.17))	`sobel_filter`
Binary thresholding (Eq. (2.18))	`threshold`
Closing holes in binary data (Eq. (2.19))	`morph_filter`
Finding connected-components and extracting the largest structure (Eq. (2.20))	`find_largest_object`
Spatial normalization and zero-padding (Eq. (2.4) and (2.21))	`zeropad`
Creating a multi-scale pyramid (Eq. (2.22))	`create_pyramid`

Table B.17.: The correspondence between mathematical equations and implemented functions for the data pre-processing and normalization stage.

Method description	Function name
Input data preparation (3D context slices) (Eq. (2.23) to (2.25))	`localize_data_prep`
Model training process	`DataLocalizator.train`
Model prediction process (Eq. (2.26))	`DataLocalizator.predict`
Creating 3D bounding boxes (Eq. (2.27) and (2.28))	`create_bbox`
Propagating 3D bounding boxes through a multi-scale pyramid (Eq. (2.29))	`scale_pyramid_bbox`
Accessing data at a specific pyramid level (Eq. (2.30))	`get_data_at_level`

Table B.18.: The correspondence between mathematical equations and implemented functions for the localization stage.

Method description	Function name
Input data preparation (the slice-wise strategy) (Eq. (2.31) to (2.33))	`seg_slicewise_prep`
Input data preparation (the patch-based strategy) (Eq. (2.31) and (2.34))	`seg_patch_prep`
Model training process (Sec. 2.6.3)	`DataSegmenter.train`
Spatial normalization of a localized structure (Eq. (2.38))	`normalize_struct`
Model prediction process (Eq. (2.39))	`DataSegmenter.predict`
Assembling image from patches (Eq. (2.40))	`combine_patches`
Obtaining 3D binary segmentation by averaging and thresholding probabilities from all orthogonal views (Eq. (2.41) and (2.42))	`average`

Table B.19.: The correspondence between mathematical equations and implemented functions for the segmentation stage.

Method description	Function name
Extracting volumetric properties (Eq. (2.44))	`extract_vol_props`
Extracting surface properties (Eq. (2.45))	`extract_surf_props`
Extracting fibers properties (Eq. (2.45))	`extract_fiber_props`

Table B.20.: The correspondence between mathematical equations and implemented functions for the morphological and statistical analysis stage.

B.5. Implementation description of fiber analysis workflow

Method description	Function name
Otsu thresholding segmentation method (Sec. 2.4)	`skimage.filters.threshold_-otsu`

Table B.21.: The descriptions of implemented functions for the fiber segmentation stage.

Method description	Function name
Fiber skeletonization (Sec. 3.1)	`skimage.morphology.skeletonize`
Fiber orientation analysis (Sec. 3.2.3)	`orientation_3d_ray_cast`
Particles counting (Sec. 3.3.2)	`object_counter`
Fiber diameter estimation (Sec. 3.3.2)	`estimate_diameter_single_run`

Table B.22.: The descriptions of implemented functions for the fiber analysis stage.

List of Figures

List of Tables

Bibliography

[1] M. W. Davidson, "Pioneers in optics: Zacharias Janssen and Johannes Kepler," *Microscopy Today*, vol. 17, no. 6, pp. 44–47, 2009.

[2] T. C. Kriss and V. M. Kriss, "History of the operating microscope: from magnifying glass to microneurosurgery," *Neurosurgery*, vol. 42, no. 4, pp. 899–907, 1998.

[3] A. Mudry, "The history of the microscope for use in ear surgery," *Otology & Neurotology*, vol. 21, no. 6, pp. 877–886, 2000.

[4] P. Willmott, *An introduction to synchrotron radiation: techniques and applications.* Chichester, West Sussex, UK: John Wiley & Sons, 2011.

[5] W. A. Kalender, *Computed tomography: fundamentals, system technology, image quality, applications.* Chichester, West Sussex, UK: John Wiley & Sons, 2011.

[6] M. Vogelgesang, T. Farago, T. F. Morgeneyer, L. Helfen, T. dos Santos Rolo, A. Myagotin, and T. Baumbach, "Real-time image-content-based beamline control for smart 4D X-ray imaging," *Journal of Synchrotron Radiation*, vol. 23, no. 5, pp. 1254–1263, 2016.

[7] A. H. Schwermann, T. dos Santos Rolo, M. S. Caterino, G. Bechly, H. Schmied, T. Baumbach, and T. van de Kamp, "Preservation of three-dimensional anatomy in phosphatized fossil arthropods enriches evolutionary inference," *eLife*, vol. 5, p. e12129, 2016.

[8] J. P. Blewett, "Synchrotron radiation–early history," *Journal of Synchrotron Radiation*, vol. 5, no. 3, pp. 135–139, 1998.

[9] D. Iwanenko and I. Pomeranchuk, "On the maximal energy attainable in a betatron," *Physical Review*, vol. 65, no. 11-12, p. 343, 1944.

[10] F. R. Elder, R. V. Langmuir, and H. C. Pollock, "Radiation from electrons accelerated in a synchrotron," *Physical Review*, vol. 74, no. 1, pp. 52–56, 1948.

[11] D. H. Tomboulian and P. L. Hartman, "Spectral and angular distribution of ultraviolet radiation from the 300-Mev Cornell synchrotron," *Physical Review*, vol. 102, no. 6, pp. 1423–1447, 1956.

[12] S. Basu and Y. Bresler, "Filtered backprojection reconstruction algorithm for tomography," *IEEE Transactions on Image Processing*, vol. 9, no. 10, pp. 1760–1773, 2000.

[13] H. Stark, J. Woods, I. Paul, and R. Hingorani, "Direct Fourier reconstruction in computer tomography," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 29, no. 2, pp. 237–245, 1981.

[14] R. Shkarin, E. Ametova, S. Chilingaryan, T. Dritschler, A. Kopmann, A. Mirone, A. Shkarin, M. Vogelgesang, and S. Tsapko, "GPU-optimized direct Fourier method for on-line tomography," *Fundamenta Informaticae*, vol. 141, no. 2-3, pp. 245–258, 2015.

[15] H. D. Litzlbauer, C. Neuhaeuser, A. Moell, S. Greschus, A. Breithecker, F. E. Franke, W. Kummer, and W. S. Rau, "Three-dimensional imaging and morphometric analysis of alveolar tissue from microfocal X-ray-computed tomography," *American Journal of Physiology-Lung Cellular and Molecular Physiology*, vol. 291, no. 3, pp. L535–L545, 2006.

[16] C. Kamenz and G. Weidemann, "Heavy metal—a contrasting substance for micro-tomographical visualization of scorpion book lungs," *Micron*, vol. 40, no. 8, pp. 911–917, 2009.

[17] A. Lareida, F. Beckmann, A. Schrott-Fischer, R. Glueckert, W. Freysinger, and B. Müller, "High-resolution X-ray tomography of the human inner ear: synchrotron radiation-based study of nerve fibre bundles, membranes and ganglion cells," *Journal of Microscopy*, vol. 234, no. 1, pp. 95–102, 2009.

[18] R. Mizutani, A. Takeuchi, K. Uesugi, S. Takekoshi, R. Y. Osamura, and Y. Suzuki, "Three-dimensional microstructural analysis of human brain tissue by using synchrotron radiation microtomographs," *Handbook on White Matter*, pp. 247–277, 2009.

[19] R. Mizutani, A. Takeuchi, R. Y. Osamura, S. Takekoshi, K. Uesugi, and Y. Suzuki, "Submicrometer tomographic resolution examined using a micro-fabricated test object," *Micron*, vol. 41, no. 1, pp. 90–95, 2010.

[20] J. Li, A. Chaudhary, S. J. Chmura, C. Pelizzari, T. Rajh, C. Wietholt, M. Kurtoglu, and B. Aydogan, "A novel functional CT contrast agent for molecular imaging of cancer," *Physics in Medicine & Biology*, vol. 55, no. 15, pp. 4389–4397, 2010.

[21] J. Nelson, X. Huang, J. Steinbrener, D. Shapiro, J. Kirz, S. Marchesini, A. M. Neiman, J. J. Turner, and C. Jacobsen, "High-resolution X-ray diffraction microscopy of specifically labeled yeast cells," *Proceedings of the National Academy of Sciences*, vol. 107, no. 16, pp. 7235–7239, 2010.

[22] H. Wang, L. Zheng, C. Peng, R. Guo, M. Shen, X. Shi, and G. Zhang, "Computed tomography imaging of cancer cells using acetylated dendrimer-entrapped gold nanoparticles," *Biomaterials*, vol. 32, no. 11, pp. 2979–2988, 2011.

[23] R. Mizutani, A. Takeuchi, K. Uesugi, S. Takekoshi, R. Y. Osamura, and Y. Suzuki, "X-ray microtomographic imaging of three-dimensional structure of soft tissues," *Tissue Engineering Part C: Methods*, vol. 14, no. 4, pp. 359–363, 2008.

[24] R. Mizutani, A. Takeuchi, K. Uesugi, M. Ohyama, S. Takekoshi, R. Y. Osamura, and Y. Suzuki, "Three-dimensional microtomographic imaging of human brain cortex," *Brain Research*, vol. 1199, pp. 53–61, 2008.

[25] H. Parameswaran, E. Bartolák-Suki, H. Hamakawa, A. Majumdar, P. G. Allen, and B. Suki, "Three-dimensional measurement of alveolar airspace volumes in normal and emphysematous lungs using micro-CT," *Journal of Applied Physiology*, vol. 107, no. 2, pp. 583–592, 2009.

[26] R. Mizutani, A. Takeuchi, K. Uesugi, S. Takekoshi, R. Y. Osamura, and Y. Suzuki, "Microtomographic analysis of neuronal circuits of human brain," *Cerebral Cortex*, vol. 20, no. 7, pp. 1739–1748, 2010.

[27] A. de Crespigny, H. Bou-Reslan, M. C. Nishimura, H. Phillips, R. A. Carano, and H. E. D'Arceuil, "3D micro-CT imaging of the postmortem brain," *Journal of Neuroscience Methods*, vol. 171, no. 2, pp. 207–213, 2008.

[28] B. D. Metscher, "Micro-CT for comparative morphology: simple staining methods allow high-contrast 3D imaging of diverse non-mineralized animal tissues," *BMC Physiology*, vol. 9, no. 1, pp. 9–11, 2009.

[29] S. J. Lee, S. Y. Jung, and S. Ahn, "Flow tracing microparticle sensors designed for enhanced X-ray contrast," *Biosensors and Bioelectronics*, vol. 25, no. 7, pp. 1571–1578, 2010.

[30] N. S. Jeffery, R. S. Stephenson, J. A. Gallagher, J. C. Jarvis, and P. G. Cox, "Micro-computed tomography with iodine staining resolves the arrangement of muscle fibres," *Journal of Biomechanics*, vol. 44, no. 1, pp. 189–192, 2011.

[31] E. Pauwels, D. Van Loo, P. Cornillie, L. Brabant, and L. Van Hoorebeke, "An exploratory study of contrast agents for soft tissue visualization by means of high resolution X-ray computed tomography imaging," *Journal of Microscopy*, vol. 250, no. 1, pp. 21–31, 2013.

[32] B. D. Metscher, "Micro-CT for developmental biology: a versatile tool for high-contrast 3D imaging at histological resolutions," *Developmental Dynamics*, vol. 238, no. 3, pp. 632–640, 2009.

[33] M. Nierenberger, Y. Rémond, S. Ahzi, and P. Choquet, "Assessing the three-dimensional collagen network in soft tissues using contrast agents and high resolution micro-CT: application to porcine iliac veins," *Comptes Rendus - Biologies*, vol. 338, no. 7, pp. 425–433, 2015.

[34] P. Cloetens, W. Ludwig, J. Baruchel, J.-P. Guigay, P. Pernot-Rejmánková, M. Salomé-Pateyron, M. Schlenker, J.-Y. Buffière, E. Maire, and G. Peix, "Hard X-ray phase imaging using simple propagation of a coherent synchrotron radiation beam," *Journal of Physics D: Applied Physics*, vol. 32, no. 10A, p. A145, 1999.

[35] J. Moosmann, R. Hofmann, and T. Baumbach, "Single-distance phase retrieval at large phase shifts," *Optics Express*, vol. 19, no. 13, pp. 12066–12073, 2011.

[36] O. V. Aslanidi, T. Nikolaidou, J. Zhao, B. H. Smaill, S. H. Gilbert, A. V. Holden, T. Lowe, P. J. Withers, R. S. Stephenson, J. C. Jarvis, *et al.*, "Application of micro-computed tomography with iodine staining to cardiac imaging, segmentation, and computational model development," *IEEE Transactions on Medical Imaging*, vol. 32, no. 1, pp. 8–17, 2012.

[37] L. Gjesteby, B. De Man, Y. Jin, H. Paganetti, J. Verburg, D. Giantsoudi, and G. Wang, "Metal artifact reduction in CT: where are we after four decades," *IEEE Access*, vol. 4, pp. 5826–5849, 2016.

[38] W. A. Kalender, R. Hebel, and J. Ebersberger, "Reduction of CT artifacts caused by metallic implants," *Radiology*, vol. 164, no. 2, pp. 576–577, 1987.

[39] C. Lemmens, D. Faul, and J. Nuyts, "Suppression of metal artifacts in CT using a reconstruction procedure that combines MAP and projection completion," *IEEE Transactions on Medical Imaging*, vol. 28, no. 2, pp. 250–260, 2008.

[40] E. Meyer, R. Raupach, M. Lell, B. Schmidt, and M. Kachelrieß, "Normalized metal artifact reduction (NMAR) in computed tomography," *Medical physics*, vol. 37, no. 10, pp. 5482–5493, 2010.

[41] M. Bal, H. Celik, K. Subramanyan, K. Eck, and L. Spies, "A radial adaptive filter for metal artifact reduction," in *Medical Imaging 2005: Image Processing*, vol. 5747, (San Diego, CA, USA), pp. 2075–2082, International Society for Optics and Photonics, 2005.

[42] O. Watzke and W. A. Kalender, "A pragmatic approach to metal artifact reduction in CT: merging of metal artifact reduced images," *European radiology*, vol. 14, no. 5, pp. 849–856, 2004.

[43] Y. Zhang and H. Yu, "Convolutional neural network based metal artifact reduction in X-ray computed tomography," *IEEE Transactions on Medical Imaging*, vol. 37, no. 6, pp. 1370–1381, 2018.

[44] C. Raven, "Numerical removal of ring artifacts in microtomography," *Review of Scientific Instruments*, vol. 69, no. 8, pp. 2978–2980, 1998.

[45] Z. Wei, S. Wiebe, and D. Chapman, "Ring artifacts removal from synchrotron CT image slices," *Journal of Instrumentation*, vol. 8, no. 06, p. C06006, 2013.

[46] E. M. A. Anas, S. Y. Lee, and M. K. Hasan, "Classification of ring artifacts for their effective removal using type adaptive correction schemes," *Computers in Biology and Medicine*, vol. 41, no. 6, pp. 390–401, 2011.

[47] B. Münch, P. Trtik, F. Marone, and M. Stampanoni, "Stripe and ring artifact removal with combined wavelet–Fourier filtering," *Optics Express*, vol. 17, no. 10, pp. 8567–8591, 2009.

[48] M. Yousuf and M. Asaduzzaman, "An efficient ring artifact reduction method based on projection data for micro-CT images," *Journal of Scientific Research*, vol. 2, no. 1, pp. 37–45, 2010.

[49] M. K. Hasan, F. Sadi, and S. Y. Lee, "Removal of ring artifacts in micro-CT imaging using iterative morphological filters," *Signal, Image and Video Processing*, vol. 6, no. 1, pp. 41–53, 2012.

[50] D. Jha, H. O. Sørensen, S. Dobberschütz, R. Feidenhans'l, and S. L. S. Stipp, "Adaptive center determination for effective suppression of ring artifacts in tomography images," *Applied Physics Letters*, vol. 105, no. 14, p. 143107, 2014.

[51] E. X. Miqueles, J. Rinkel, F. O'Dowd, and J. S. Bermúdez, "Generalized Titarenko's algorithm for ring artefacts reduction," *Journal of Synchrotron Radiation*, vol. 21, no. 6, pp. 1333–1346, 2014.

[52] S. Yao, Y. Zong, J. Fan, Z. Sun, J. Zhang, and H. Jiang, "Synchrotron X-ray microtomography with improved image quality by ring artifacts correction for structural analysis of insects," *Microscopy and Microanalysis*, vol. 23, no. 5, pp. 938–944, 2017.

[53] P. Paleo and A. Mirone, "Ring artifacts correction in compressed sensing tomographic reconstruction," *Journal of Synchrotron Radiation*, vol. 22, no. 5, pp. 1268–1278, 2015.

[54] D. M. Pelt and D. Y. Parkinson, "Ring artifact reduction in synchrotron X-ray tomography through helical acquisition," *Measurement Science and Technology*, vol. 29, no. 3, p. 034002, 2018.

[55] J. Sijbers and A. Postnov, "Reduction of ring artefacts in high resolution micro-CT reconstructions," *Physics in Medicine & Biology*, vol. 49, no. 14, p. N247, 2004.

[56] Y. Kyriakou, D. Prell, and W. A. Kalender, "Ring artifact correction for high-resolution micro CT," *Physics in Medicine & Biology*, vol. 54, no. 17, p. N385, 2009.

[57] F. E. Boas and D. Fleischmann, "CT artifacts: causes and reduction techniques," *Imaging in Medicine*, vol. 4, no. 2, pp. 229–240, 2012.

[58] A. Pacureanu, M. Langer, E. Boller, P. Tafforeau, and F. Peyrin, "Nanoscale imaging of the bone cell network with synchrotron X-ray tomography: optimization of acquisition setup," *Medical Physics*, vol. 39, no. 4, pp. 2229–2238, 2012.

[59] I. Dimitrovski, D. Kocev, I. Kitanovski, S. Loskovska, and S. Džeroski, "Improved medical image modality classification using a combination of visual and textual features," *Computerized Medical Imaging and Graphics*, vol. 39, pp. 14–26, 2015.

[60] D. H. Hubel and T. N. Wiesel, "Receptive fields of single neurones in the cat's striate cortex," *The Journal of Physiology*, vol. 148, no. 3, pp. 574–591, 1959.

[61] D. H. Hubel and T. N. Wiesel, "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex," *The Journal of Physiology*, vol. 160, no. 1, pp. 106–154, 1962.

[62] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[63] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural Computation*, vol. 1, no. 4, pp. 541–551, 1989.

[64] D. Yu, H. Wang, P. Chen, and Z. Wei, "Mixed pooling for convolutional neural networks," in *International Conference on Rough Sets and Knowledge Technology*, (Shanghai, China), pp. 364–375, Springer, 2014.

[65] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, (Haifa, Israel), pp. 807–814, Omnipress, 2010.

[66] M. Ranzato, F. J. Huang, Y.-L. Boureau, and Y. LeCun, "Unsupervised learning of invariant feature hierarchies with applications to object recognition," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, (Minneapolis, MN, USA), pp. 1–8, IEEE, 2007.

[67] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[68] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, pp. 1–14, 2014.

[69] E. A. Susaki and H. R. Ueda, "Whole-body and whole-organ clearing and imaging techniques with single-cell resolution: toward organism-level systems biology in mammals," *Cell Chemical Biology*, vol. 23, no. 1, pp. 137–157, 2016.

[70] S. D. Brown, C. C. Holmes, A.-M. Mallon, T. F. Meehan, D. Smedley, and S. Wells, "High-throughput mouse phenomics for characterizing mammalian gene function," *Nature Reviews Genetics*, vol. 19, no. 6, pp. 357–370, 2018.

[71] J. Stegmaier, B. Schott, E. Hübner, M. Traub, M. Shahid, M. Takamiya, A. Kobitski, V. Hartmann, R. Stotzka, J. van Wezel, *et al.*, "Automation strategies for large-scale 3D image analysis," *At-Automatisierungstechnik*, vol. 64, no. 7, pp. 555–566, 2016.

[72] S. J. Schambach, S. Bag, L. Schilling, C. Groden, and M. A. Brockmann, "Application of micro-CT in small animal imaging," *Methods*, vol. 50, no. 1, pp. 2–13, 2010.

[73] D. Clark and C. Badea, "Micro-CT of rodents: state-of-the-art and future perspectives," *Physica Medica*, vol. 30, no. 6, pp. 619–634, 2014.

[74] M. D. Wong, Y. Maezawa, J. P. Lerch, and R. M. Henkelman, "Automated pipeline for anatomical phenotyping of mouse embryos using micro-CT," *Development*, vol. 141, no. 12, pp. 2533–2541, 2014.

[75] J. M. Brown, N. R. Horner, T. N. Lawson, T. Fiegel, S. Greenaway, H. Morgan, N. Ring, L. Santos, D. Sneddon, L. Teboul, *et al.*, "A bioimage informatics platform for high-throughput embryo phenotyping," *Briefings in Bioinformatics*, vol. 19, no. 1, pp. 41–51, 2018.

[76] Z. Xie, A. Kitamoto, M. Tamura, T. Shiroishi, and D. Gillies, "High-throughput mouse phenotyping using non-rigid registration and robust principal component analysis," in *Medical Imaging 2016: Image Processing*, vol. 9784, (San Diego, CA, USA), p. 978415, International Society for Optics and Photonics, 2016.

[77] E. Rebollo, K. Karkali, F. Mangione, and E. Martín-Blanco, "Live imaging in Drosophila: the optical and genetic toolkits," *Methods*, vol. 68, no. 1, pp. 48–59, 2014.

[78] J. Stegmaier, F. Amat, W. C. Lemon, K. McDole, Y. Wan, G. Teodoro, R. Mikut, and P. J. Keller, "Real-time three-dimensional cell segmentation in large-scale microscopy data of developing embryos," *Developmental Cell*, vol. 36, no. 2, pp. 225–240, 2016.

[79] H. Dankert, L. Wang, E. D. Hoopfer, D. J. Anderson, and P. Perona, "Automated monitoring and analysis of social behavior in Drosophila," *Nature Methods*, vol. 6, no. 4, p. 297, 2009.

[80] H. Peng, P. Chung, F. Long, L. Qu, A. Jenett, A. M. Seeds, E. W. Myers, and J. H. Simpson, "BrainAligner: 3D registration atlases of Drosophila brains," *Nature Methods*, vol. 8, no. 6, pp. 493–498, 2011.

[81] K. Panser, L. Tirian, F. Schulze, S. Villalba, G. S. Jefferis, K. Bühler, and A. D. Straw, "Automatic segmentation of Drosophila neural compartments using GAL4 expression data reveals novel visual pathways," *Current Biology*, vol. 26, no. 15, pp. 1943–1954, 2016.

[82] F. Piccinini, E. Lucarelli, A. Gherardi, and A. Bevilacqua, "Multi-image based method to correct vignetting effect in light microscopy images," *Journal of Microscopy*, vol. 248, no. 1, pp. 6–22, 2012.

[83] A. Bevilacqua, F. Piccinini, and A. Gherardi, "Vignetting correction by exploiting an optical microscopy image sequence," in *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, (Boston, MA, USA), pp. 6166–6169, IEEE, 2011.

[84] S. C. Lee and P. Bajcsy, "Intensity correction of fluorescent confocal laser scanning microscope images by mean-weight filtering," *Journal of Microscopy*, vol. 221, no. 2, pp. 122–136, 2006.

[85] S. Ramani, C. Vonesch, and M. Unser, "Deconvolution of 3D fluorescence micrographs with automatic risk minimization," in *2008 5th IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, (Paris, France), pp. 732–735, IEEE, 2008.

[86] S. Li, B. Yang, and J. Hu, "Performance comparison of different multi-resolution transforms for image fusion," *Information Fusion*, vol. 12, no. 2, pp. 74–84, 2011.

[87] Q. Chen and D. Wu, "Image denoising by bounded block matching and 3D filtering," *Signal Processing*, vol. 90, no. 9, pp. 2778–2783, 2010.

[88] V. Van Nieuwenhove, J. De Beenhouwer, F. De Carlo, L. Mancini, F. Marone, and J. Sijbers, "Dynamic intensity normalization using eigen flat fields in X-ray imaging," *Optics Express*, vol. 23, no. 21, pp. 27975–27989, 2015.

[89] J. A. Seibert, J. M. Boone, and K. K. Lindfors, "Flat-field correction technique for digital detectors," in *Medical Imaging 1998: Physics of Medical Imaging*, vol. 3336, (San Diego, CA, USA), pp. 348–354, International Society for Optics and Photonics, 1998.

[90] L. Ritschl, F. Bergner, C. Fleischmann, and M. Kachelrieß, "Improved total variation-based CT image reconstruction applied to clinical data," *Physics in Medicine & Biology*, vol. 56, no. 6, p. 1545, 2011.

[91] A. Mirone, E. Brun, E. Gouillart, P. Tafforeau, and J. Kieffer, "The PyHST2 hybrid distributed code for high speed tomographic reconstruction with iterative reconstruction and a priori knowledge capabilities," *Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms*, vol. 324, pp. 41–48, 2014.

[92] K. Akila, L. Jayashree, and A. Vasuki, "Mammographic image enhancement using indirect contrast enhancement techniques–a comparative study," *Procedia Computer Science*, vol. 47, pp. 255–261, 2015.

[93] A. Buades, B. Coll, and J.-M. Morel, "A review of image denoising algorithms, with a new one," *Multiscale Modeling & Simulation*, vol. 4, no. 2, pp. 490–530, 2005.

[94] D. Kostadin, F. Alessandro, and E. Karen, "Video denoising by sparse 3D transform-domain collaborative filtering," in *European Signal Processing Conference*, vol. 149, (Tampere, Finland), p. 2, IEEE, 2007.

[95] S. M. Pizer, E. P. Amburn, J. D. Austin, R. Cromartie, A. Geselowitz, T. Greer, B. ter Haar Romeny, J. B. Zimmerman, and K. Zuiderveld, "Adaptive histogram equalization and its variations," *Computer Vision, Graphics, and Image Processing*, vol. 39, no. 3, pp. 355–368, 1987.

[96] J. A. Stark, "Adaptive image contrast enhancement using generalizations of histogram equalization," *IEEE Transactions on Image Processing*, vol. 9, no. 5, pp. 889–896, 2000.

[97] V. Magudeeswaran and J. F. Singh, "Contrast limited fuzzy adaptive histogram equalization for enhancement of brain images," *International Journal of Imaging Systems and Technology*, vol. 27, no. 1, pp. 98–103, 2017.

[98] E. H. Adelson, C. H. Anderson, J. R. Bergen, P. J. Burt, and J. M. Ogden, "Pyramid methods in image processing," *RCA Engineer*, vol. 29, no. 6, pp. 33–41, 1984.

[99] R. Szeliski, "Image alignment and stitching," in *Handbook of Mathematical Models in Computer Vision*, pp. 273–292, Springer, 2006.

[100] P. Thévenaz and M. Unser, "User-friendly semiautomated assembly of accurate image mosaics in microscopy," *Microscopy Research and Technique*, vol. 70, no. 2, pp. 135–146, 2007.

[101] P. Bajcsy, S.-C. Lee, A. Lin, and R. Folberg, "Three-dimensional volume reconstruction of extracellular matrix proteins in uveal melanoma from fluorescent confocal laser scanning microscope images," *Journal of Microscopy*, vol. 221, no. 1, pp. 30–45, 2006.

[102] S. K. Chow, H. Hakozaki, D. L. Price, N. A. MacLean, T. J. Deerinck, J. C. Bouwer, M. E. Martone, S. T. Peltier, and M. H. Ellisman, "Automated microscopy system for mosaic acquisition and processing," *Journal of Microscopy*, vol. 222, no. 2, pp. 76–84, 2006.

[103] D. Ni, Y. P. Chui, Y. Qu, X. Yang, J. Qin, T.-T. Wong, S. S. Ho, and P. A. Heng, "Reconstruction of volumetric ultrasound panorama based on improved 3D SIFT," *Computerized Medical Imaging and Graphics*, vol. 33, no. 7, pp. 559–566, 2009.

[104] G. T. Flitton, T. P. Breckon, and N. M. Bouallagu, "Object recognition using 3D SIFT in complex CT volumes," in *Proceedings of the British Machine Vision Conference 2010*, no. 1, (Aberystwyth, Wales, UK), pp. 1–12, British Machine Vision Association, 2010.

[105] H. Goncalves, L. Corte-Real, and J. A. Goncalves, "Automatic image registration through image segmentation and SIFT," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 49, no. 7, pp. 2589–2600, 2011.

[106] B. Rister, M. A. Horowitz, and D. L. Rubin, "Volumetric image registration from invariant keypoints," *IEEE Transactions on Image Processing*, vol. 26, no. 10, pp. 4900–4910, 2017.

[107] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[108] P. Lukashevich, B. Zalesky, and S. Ablameyko, "Medical image registration based on SURF detector," *Pattern Recognition and Image Analysis*, vol. 21, no. 3, p. 519, 2011.

[109] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide-baseline stereo from maximally stable extremal regions," *Image and Vision Computing*, vol. 22, no. 10, pp. 761–767, 2004.

[110] D. Dean, "Statistical shape analysis," *Journal of Human Evolution*, vol. 38, pp. 455–457, 2000.

[111] J. M. Fitzpatrick, J. B. West, and C. R. Maurer, "Predicting error in rigid-body point-based registration," *IEEE Transactions on Medical Imaging*, vol. 17, no. 5, pp. 694–702, 1998.

[112] E. Cuchet, J. Knoplioch, D. Dormont, and C. Marsault, "Registration in neurosurgery and neuroradiotherapy applications," *Journal of Image Guided Surgery*, vol. 1, no. 4, pp. 198–207, 1995.

[113] J. Declerck, J. Feldmar, M. L. Goris, and F. Betting, "Automatic registration and alignment on a template of cardiac stress and rest reoriented SPECT images," *IEEE Transactions on Medical Imaging*, vol. 16, no. 6, pp. 727–737, 1997.

[114] C. R. Maurer, R. J. Maciunas, and J. M. Fitzpatrick, "Registration of head CT images to physical space using a weighted combination of points and surfaces," *IEEE Transactions on Medical Imaging*, vol. 17, no. 5, pp. 753–761, 1998.

[115] R. P. Woods, S. T. Grafton, C. J. Holmes, S. R. Cherry, and J. C. Mazziotta, "Automated image registration: I. general methods and intrasubject, intramodality validation," *Journal of Computer Assisted Tomography*, vol. 22, no. 1, pp. 139–152, 1998.

[116] J. Ashburner and K. J. Friston, "Nonlinear spatial normalization using basis functions," *Human Brain Mapping*, vol. 7, no. 4, pp. 254–266, 1999.

[117] A. Roche, G. Malandain, X. Pennec, and N. Ayache, "The correlation ratio as a new similarity measure for multimodal image registration," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, (Cambridge, MA, USA), pp. 1115–1124, Springer, 1998.

[118] G. Hermosillo, C. Chefd'Hotel, and O. Faugeras, "Variational methods for multimodal image matching," *International Journal of Computer Vision*, vol. 50, no. 3, pp. 329–343, 2002.

[119] J. Orchard, "Globally optimal multimodal rigid registration: an analytic solution using edge information," in *2007 IEEE International Conference on Image Processing*, vol. 1, (San Antonio, TX, USA), pp. I–485, IEEE, 2007.

[120] W. M. Wells III, P. Viola, H. Atsumi, S. Nakajima, and R. Kikinis, "Multimodal volume registration by maximization of mutual information," *Medical Image Analysis*, vol. 1, no. 1, pp. 35–51, 1996.

[121] A. Collignon, F. Maes, D. Delaere, D. Vandermeulen, P. Suetens, and G. Marchal, "Automated multi-modality image registration based on information theory," in *Information Processing in Medical Imaging*, vol. 3, (Dordrecht, Netherlands), pp. 263–274, Springer, 1995.

[122] A. Collignon, *Multi-modality medical image registration by maximization of mutual information*. PhD thesis, KU Leuven, 1998.

[123] J. P. Pluim, J. A. Maintz, and M. A. Viergever, "Mutual-information-based registration of medical images: a survey," *IEEE Transactions on Medical Imaging*, vol. 22, no. 8, pp. 986–1004, 2003.

[124] F. Maes, D. Vandermeulen, and P. Suetens, "Medical image registration using mutual information," *Proceedings of the IEEE*, vol. 91, no. 10, pp. 1699–1722, 2003.

[125] J. P. Pluim, J. A. Maintz, and M. A. Viergever, "F-information measures in medical image registration," *IEEE Transactions on Medical Imaging*, vol. 23, no. 12, pp. 1508–1516, 2004.

[126] Y. He, A. B. Hamza, and H. Krim, "A generalized divergence measure for robust image registration," *IEEE Transactions on Signal Processing*, vol. 51, no. 5, pp. 1211–1220, 2003.

[127] C. Tsallis, "Possible generalization of Boltzmann-Gibbs statistics," *Journal of Statistical Physics*, vol. 52, no. 1-2, pp. 479–487, 1988.

[128] S. Sun, L. Zhang, and C. Guo, "Medical image registration by minimizing divergence measure based on Tsallis entropy," *International Journal of Biological and Medical Sciences*, vol. 2, no. 2, pp. 75–80, 2007.

[129] M. Khader and A. B. Hamza, "An entropy-based technique for nonrigid medical image alignment," in *International Workshop on Combinatorial Image Analysis*, (Madrid, Spain), pp. 444–455, Springer, 2011.

[130] M. P. Wachowiak, R. Smolíková, and T. M. Peters, "Multiresolution biomedical image registration using generalized information measures," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, (Montreal, QC, Canada), pp. 846–853, Springer, 2003.

[131] S. Seifert, A. Barbu, S. K. Zhou, D. Liu, J. Feulner, M. Huber, M. Suehling, A. Cavallaro, and D. Comaniciu, "Hierarchical parsing and semantic navigation of full body CT data," in *Medical Imaging 2009: Image Processing*, vol. 7259, (Lake Buena Vista, FL, USA), p. 725902, International Society for Optics and Photonics, 2009.

[132] R. Gauriau, R. Cuingnet, D. Lesage, and I. Bloch, "Multi-organ localization with cascaded global-to-local regression and shape prior," *Medical Image Analysis*, vol. 23, no. 1, pp. 70–83, 2015.

[133] T. Okada, M. G. Linguraru, M. Hori, R. M. Summers, N. Tomiyama, and Y. Sato, "Abdominal multi-organ segmentation from CT images using conditional shape–location and unsupervised intensity priors," *Medical Image Analysis*, vol. 26, no. 1, pp. 1–18, 2015.

[134] M. Zreik, T. Leiner, B. D. De Vos, R. W. van Hamersvelt, M. A. Viergever, and I. Išgum, "Automatic segmentation of the left ventricle in cardiac CT angiography using convolutional neural networks," in *2016 IEEE 13th International Symposium on Biomedical Imaging*, (Prague, Czech Republic), pp. 40–43, IEEE, 2016.

[135] J. M. Wolterink, T. Leiner, B. D. de Vos, R. W. van Hamersvelt, M. A. Viergever, and I. Išgum, "Automatic coronary artery calcium scoring in cardiac CT angiography using paired convolutional neural networks," *Medical Image Analysis*, vol. 34, pp. 123–136, 2016.

[136] N. Lessmann, I. Išgum, A. A. Setio, B. D. de Vos, F. Ciompi, P. A. de Jong, M. Oudkerk, P. T. M. Willem, M. A. Viergever, and B. van Ginneken, "Deep convolutional neural networks for automatic coronary calcium scoring in a screening study with low-dose chest CT," in *Medical Imaging 2016: Computer-Aided Diagnosis*, vol. 9785, (San Diego, CA, USA), p. 978511, International Society for Optics and Photonics, 2016.

[137] H. R. Roth, C. T. Lee, H.-C. Shin, A. Seff, L. Kim, J. Yao, L. Lu, and R. M. Summers, "Anatomy-specific classification of medical images using deep convolutional nets," in *2015 IEEE 12th International Symposium on Biomedical Imaging*, (New York, NY, USA), pp. 101–104, IEEE, 2015.

[138] A. Alzu'bi, A. Amira, and N. Ramzan, "Semantic content-based image retrieval: a comprehensive study," *Journal of Visual Communication and Image Representation*, vol. 32, pp. 20–54, 2015.

[139] X. Zhang, W. Liu, M. Dundar, S. Badve, and S. Zhang, "Towards large-scale histopathological image analysis: hashing-based image retrieval," *IEEE Transactions on Medical Imaging*, vol. 34, no. 2, pp. 496–506, 2014.

[140] R. Murthy, R. Nunez, J. Szklaruk, W. Erwin, D. C. Madoff, S. Gupta, K. Ahrar, M. J. Wallace, A. Cohen, D. M. Coldwell, *et al.*, "Yttrium-90 microsphere therapy for hepatic malignancy: devices, indications, technical considerations, and potential complications," *Radiographics*, vol. 25, no. 1, pp. S41–S55, 2005.

[141] X. Liu, Q. Song, P. Mendonca, X. Tao, and R. Bhotika, "Organ labeling using anatomical model-driven global optimization," in *2011 IEEE First International Conference on Healthcare Informatics, Imaging and Systems Biology*, (San Jose, CA, USA), pp. 338–345, IEEE, 2011.

[142] G. Fouquier, J. Atif, and I. Bloch, "Sequential model-based segmentation and recognition of image structures driven by visual features and spatial relations," *Computer Vision and Image Understanding*, vol. 116, no. 1, pp. 146–165, 2012.

[143] O. Nempont, J. Atif, and I. Bloch, "A constraint propagation approach to structural model based image segmentation and recognition," *Information Sciences*, vol. 246, pp. 1–27, 2013.

[144] F. van der Lijn, T. Den Heijer, M. M. Breteler, and W. J. Niessen, "Hippocampus segmentation in MR images using atlas registration, voxel classification, and graph cuts," *NeuroImage*, vol. 43, no. 4, pp. 708–720, 2008.

[145] I. Isgum, M. Staring, A. Rutten, M. Prokop, M. A. Viergever, and B. Van Ginneken, "Multi-atlas-based segmentation with local decision fusion–application to cardiac and aortic segmentation in CT scans," *IEEE Transactions on Medical Imaging*, vol. 28, no. 7, pp. 1000–1010, 2009.

[146] H. Kirişli, M. Schaap, S. Klein, S.-L. Papadopoulou, M. Bonardi, C.-H. Chen, A. C. Weustink, N. R. Mollet, E.-J. Vonken, R. J. van der Geest, *et al.*, "Evaluation of a multi-atlas based method for segmentation of cardiac CTA data: a large-scale, multicenter, and multivendor study," *Medical Physics*, vol. 37, no. 12, pp. 6279–6291, 2010.

[147] S. R. Ranjan, "Organ localization through anatomy-aware non-rigid registration with atlas," in *2011 IEEE Applied Imagery Pattern Recognition Workshop*, (Washington, DC, USA), pp. 1–5, IEEE, 2011.

[148] J. E. Iglesias and M. R. Sabuncu, "Multi-atlas segmentation of biomedical images: a survey," *Medical Image Analysis*, vol. 24, no. 1, pp. 205–219, 2015.

[149] T. Rohlfing, R. Brandt, R. Menzel, and C. R. Maurer Jr, "Evaluation of atlas selection strategies for atlas-based image segmentation with application to confocal microscopy images of bee brains," *NeuroImage*, vol. 21, no. 4, pp. 1428–1442, 2004.

[150] A. J. Asman, F. W. Bryan, S. A. Smith, D. S. Reich, and B. A. Landman, "Groupwise multi-atlas segmentation of the spinal cord's internal structure," *Medical Image Analysis*, vol. 18, no. 3, pp. 460–471, 2014.

[151] G. Sanroma, G. Wu, Y. Gao, and D. Shen, "Learning to rank atlases for multiple-atlas segmentation," *IEEE Transactions on Medical Imaging*, vol. 33, no. 10, pp. 1939–1953, 2014.

[152] Y. Zheng, A. Barbu, B. Georgescu, M. Scheuering, and D. Comaniciu, "Fast automatic heart chamber segmentation from 3D CT data using marginal space learning and steerable features," in *2007 IEEE 11th International Conference on Computer Vision*, (Rio de Janeiro, Brazil), pp. 1–8, IEEE, 2007.

[153] Y. Zheng, B. Georgescu, and D. Comaniciu, "Marginal space learning for efficient detection of 2D/3D anatomical structures in medical images," in *International Conference on Information Processing in Medical Imaging*, (Williamsburg, VA, USA), pp. 411–422, Springer, 2009.

[154] A. Criminisi, D. Robertson, E. Konukoglu, J. Shotton, S. Pathak, S. White, and K. Siddiqui, "Regression forests for efficient anatomy detection and localization in computed tomography scans," *Medical Image Analysis*, vol. 17, no. 8, pp. 1293–1303, 2013.

[155] J. Shotton, J. Winn, C. Rother, and A. Criminisi, "TextonBoost for image understanding: multi-class object recognition and segmentation by jointly modeling texture, layout, and context," *International Journal of Computer Vision*, vol. 81, no. 1, pp. 2–23, 2009.

[156] X. Zhou, S. Wang, H. Chen, T. Hara, R. Yokoyama, M. Kanematsu, and H. Fujita, "Automatic localization of solid organs on 3D CT images by a collaborative majority voting decision based on ensemble learning," *Computerized Medical Imaging and Graphics*, vol. 36, no. 4, pp. 304–313, 2012.

[157] H.-C. Shin, M. R. Orton, D. J. Collins, S. J. Doran, and M. O. Leach, "Stacked autoencoders for unsupervised feature learning and multiple organ detection in a pilot study using 4D patient data," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1930–1943, 2012.

[158] Z. Yan, Y. Zhan, Z. Peng, S. Liao, Y. Shinagawa, D. N. Metaxas, and X. S. Zhou, "Bodypart recognition using multi-stage deep learning," in *International Conference on Information Processing in Medical Imaging*, (Aird of Sleat, Scotland, UK), pp. 449–461, Springer, 2015.

[159] B. D. de Vos, J. M. Wolterink, P. A. de Jong, T. Leiner, M. A. Viergever, and I. Išgum, "ConvNet-based localization of anatomical structures in 3-D medical images," *IEEE Transactions on Medical Imaging*, vol. 36, no. 7, pp. 1470–1481, 2017.

[160] M. A. Hussain, A. Amir-Khalili, G. Hamarneh, and R. Abugharbieh, "Segmentation-free kidney localization and volume estimation using aggregated orthogonal decision CNNs," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, (Quebec City, QC, Canada), pp. 612–620, Springer, 2017.

[161] G. E. Humpire-Mamani, A. A. A. Setio, B. van Ginneken, and C. Jacobs, "Efficient organ localization using multi-label convolutional neural networks in thorax-abdomen CT scans," *Physics in Medicine & Biology*, vol. 63, no. 8, p. 085003, 2018.

[162] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham, "Active shape models–their training and application," *Computer Vision and Image Understanding*, vol. 61, no. 1, pp. 38–59, 1995.

[163] L. Gao, D. G. Heath, and E. K. Fishman, "Abdominal image segmentation using three-dimensional deformable models," *Investigative Radiology*, vol. 33, no. 6, pp. 348–355, 1998.

[164] T. Heimann, I. Wolf, and H.-P. Meinzer, "Active shape models for a fully automated 3D segmentation of the liver–an evaluation on clinical data," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, vol. 9, (Copenhagen, Denmark), pp. 41–48, Springer, 2006.

[165] A. Shimizu, T. Kimoto, H. Kobatake, S. Nawano, and K. Shinozaki, "Automated pancreas segmentation from three-dimensional contrast-enhanced computed tomography," *International Journal of Computer Assisted Radiology and Surgery*, vol. 5, no. 1, p. 85, 2010.

[166] H. Park, P. H. Bland, and C. R. Meyer, "Construction of an abdominal probabilistic atlas and its application in segmentation," *IEEE Transactions on Medical Imaging*, vol. 22, no. 4, pp. 483–492, 2003.

[167] J. J. Cerrolaza, A. Villanueva, M. Reyes, R. Cabeza, M. A. G. Ballester, and M. G. Linguraru, "Generalized multiresolution hierarchical shape models via automatic landmark clusterization," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, (Boston, MA, USA), pp. 1–8, Springer, 2014.

[168] L. Wang, K. Lekadir, E.-H. Ismail, M. Yacoub, and G.-Z. Yang, "Subject specific shape modeling with incremental mixture models," in *International Workshop on Medical Imaging and Virtual Reality*, (Beijing, China), pp. 21–30, Springer, 2010.

[169] C. Chu, M. Oda, T. Kitasaka, K. Misawa, M. Fujiwara, Y. Hayashi, Y. Nimura, D. Rueckert, and K. Mori, "Multi-organ segmentation based on spatially-divided probabilistic atlas from 3D abdominal CT images," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, (Nagoya, Japan), pp. 165–172, Springer, 2013.

[170] R. Wolz, C. Chu, K. Misawa, M. Fujiwara, K. Mori, and D. Rueckert, "Automated abdominal multi-organ segmentation with subject-specific atlas generation," *IEEE Transactions on Medical Imaging*, vol. 32, no. 9, pp. 1723–1730, 2013.

[171] M. G. Linguraru, J. K. Sandberg, Z. Li, F. Shah, and R. M. Summers, "Automated segmentation and quantification of liver and spleen from CT images using normalized probabilistic atlases and enhancement estimation," *Medical Physics*, vol. 37, no. 2, pp. 771–783, 2010.

[172] M. G. Linguraru, J. A. Pura, V. Pamulapati, and R. M. Summers, "Statistical 4D graphs for multi-organ abdominal segmentation from multiphase CT," *Medical Image Analysis*, vol. 16, no. 4, pp. 904–914, 2012.

[173] M. Oda, T. Nakaoka, T. Kitasaka, K. Furukawa, K. Misawa, M. Fujiwara, and K. Mori, "Organ segmentation from 3D abdominal CT images based on atlas selection and graph cut," in *International MICCAI Workshop on Computational and Clinical Challenges in Abdominal Imaging*, (Toronto, ON, Canada), pp. 181–188, Springer, 2011.

[174] T. Okada, K. Yokota, M. Hori, M. Nakamoto, H. Nakamura, and Y. Sato, "Construction of hierarchical multi-organ statistical atlases and their application to multi-organ segmentation from CT images," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, (New York, NY, USA), pp. 502–509, Springer, 2008.

[175] A. Shimizu, K. Nakagomi, T. Narihira, H. Kobatake, S. Nawano, K. Shinozaki, K. Ishizu, and K. Togashi, "Automated segmentation of 3D CT images based on statistical atlas and graph cuts," in *International MICCAI Workshop on Medical Computer Vision*, vol. 6533, (Beijing, China), pp. 214–223, Springer, 2010.

[176] Z. Wang, K. K. Bhatia, B. Glocker, A. Marvao, T. Dawes, K. Misawa, K. Mori, and D. Rueckert, "Geodesic patch-based segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, (Boston, MA, USA), pp. 666–673, Springer, 2014.

[177] J. Gehrig, M. Reischl, É. Kalmár, M. Ferg, Y. Hadzhiev, A. Zaucker, C. Song, S. Schindler, U. Liebel, and F. Müller, "Automated high-throughput mapping of promoter-enhancer interactions in zebrafish embryos," *Nature Methods*, vol. 6, no. 12, p. 911, 2009.

[178] P. Aljabar, R. A. Heckemann, A. Hammers, J. V. Hajnal, and D. Rueckert, "Multi-atlas based segmentation of brain images: atlas selection and its effect on accuracy," *NeuroImage*, vol. 46, no. 3, pp. 726–738, 2009.

[179] Y. Cao, Y. Yuan, X. Li, B. Turkbey, P. L. Choyke, and P. Yan, "Segmenting images by combining selected atlases on manifold," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, (Toronto, ON, Canada), pp. 272–279, Springer, 2011.

[180] R. Wolz, P. Aljabar, J. V. Hajnal, A. Hammers, D. Rueckert, A. D. N. Initiative, *et al.*, "LEAP: learning embeddings for atlas propagation," *NeuroImage*, vol. 49, no. 2, pp. 1316–1325, 2010.

[181] E. M. van Rikxoort, I. Isgum, Y. Arzhaeva, M. Staring, S. Klein, M. A. Viergever, J. P. Pluim, and B. van Ginneken, "Adaptive local multi-atlas segmentation: application to the heart and the caudate nucleus," *Medical Image Analysis*, vol. 14, no. 1, pp. 39–49, 2010.

[182] F. Shi, P.-T. Yap, Y. Fan, J. H. Gilmore, W. Lin, and D. Shen, "Construction of multi-region-multi-reference atlases for neonatal brain MRI segmentation," *NeuroImage*, vol. 51, no. 2, pp. 684–693, 2010.

[183] T. Tong, R. Wolz, P. Coupé, J. V. Hajnal, D. Rueckert, A. D. N. Initiative, *et al.*, "Segmentation of MR images via discriminative dictionary learning and sparse coding: application to hippocampus labeling," *NeuroImage*, vol. 76, pp. 11–23, 2013.

[184] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[185] P. N. Yianilos, "Data structures and algorithms for nearest neighbor search in general metric spaces," in *Proceedings of the 4th Annual Acm-Siam Symposium on Discrete Algorithms*, vol. 93, (Philadelphia, PA, USA), pp. 311–21, Society for Industrial and Applied Mathematics, 1993.

[186] M. P. Heinrich and M. Blendowski, "Multi-organ segmentation using vantage point forests and binary context features," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, (Athens, Greece), pp. 598–606, Springer, 2016.

[187] A. Criminisi, J. Shotton, E. Konukoglu, *et al.*, "Decision forests: a unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning," *Foundations and Trends in Computer Graphics and Vision*, vol. 7, no. 2–3, pp. 81–227, 2012.

[188] C. Wachinger, M. Toews, G. Langs, W. Wells, and P. Golland, "Keypoint transfer segmentation," in *International Conference on Information Processing in Medical Imaging*, (Sabhal Mor Ostaig, Isle of Skye, UK), pp. 233–245, Springer, 2015.

[189] M. Yaqub, M. K. Javaid, C. Cooper, and J. A. Noble, "Investigation of the role of feature selection and weighted voting in random forests for 3-D volumetric segmentation," *IEEE Transactions on Medical Imaging*, vol. 33, no. 2, pp. 258–271, 2013.

[190] C. Jin, F. Shi, D. Xiang, L. Zhang, and X. Chen, "Fast segmentation of kidney components using random forests and ferns," *Medical Physics*, vol. 44, no. 12, pp. 6353–6363, 2017.

[191] D. Richmond, D. Kainmueller, B. Glocker, C. Rother, and G. Myers, "Uncertainty-driven forest predictors for vertebra localization and segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, (Munich, Germany), pp. 653–660, Springer, 2015.

[192] M. Bieth, E. Alberts, M. Schwaiger, and B. Menze, "From large to small organ segmentation in CT using regional context," in *International Workshop on Machine Learning in Medical Imaging*, (Quebec City, QC, Canada), pp. 1–9, Springer, 2017.

[193] V. Zografos, A. Valentinitsch, M. Rempfler, F. Tombari, and B. Menze, "Hierarchical multi-organ segmentation without registration in 3D abdominal CT images," in *International MICCAI Workshop on Medical Computer Vision*, (Munich, Germany), pp. 37–46, Springer, 2015.

[194] B. Glocker, O. Pauly, E. Konukoglu, and A. Criminisi, "Joint classification-regression forests for spatially structured multi-object segmentation," in *European Conference on Computer Vision*, (Florence, Italy), pp. 870–881, Springer, 2012.

[195] A. Montillo, J. Tu, J. Shotton, J. Winn, J. E. Iglesias, D. N. Metaxas, and A. Criminisi, "Entanglement and differentiable information gain maximization," in *Decision Forests for Computer Vision and Medical Image Analysis*, pp. 273–293, Springer, 2013.

[196] F. Xing, Y. Xie, and L. Yang, "An automatic learning-based framework for robust nucleus segmentation," *IEEE Transactions on Medical Imaging*, vol. 35, no. 2, pp. 550–566, 2015.

[197] H. Chen, Y. Zheng, J.-H. Park, P.-A. Heng, and S. K. Zhou, "Iterative multi-domain regularized deep learning for anatomical structure detection and segmentation from ultrasound images," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, (Athens, Greece), pp. 487–495, Springer, 2016.

[198] N. Dhungel, G. Carneiro, and A. P. Bradley, "Deep learning and structured prediction for the segmentation of mass in mammograms," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, (Munich, Germany,), pp. 605–612, Springer, 2015.

[199] D. Ciresan, A. Giusti, L. M. Gambardella, and J. Schmidhuber, "Deep neural networks segment neuronal membranes in electron microscopy images," in *Advances in Neural Information Processing Systems*, (Harrah's Lake Tahoe, NV, USA), pp. 2843–2851, NIPS, 2012.

[200] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 39, (San Juan, Puerto Rico, USA), pp. 3431–3440, IEEE, 2015.

[201] K. Sirinukunwattana, S. E. A. Raza, Y.-W. Tsang, D. R. Snead, I. A. Cree, and N. M. Rajpoot, "Locality sensitive deep learning for detection and classification of nuclei in routine colon cancer histology images," *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, pp. 1196–1206, 2016.

[202] L. Yu, H. Chen, Q. Dou, J. Qin, and P.-A. Heng, "Automated melanoma recognition in dermoscopy images via very deep residual networks," *IEEE Transactions on Medical Imaging*, vol. 36, no. 4, pp. 994–1004, 2016.

[203] A. Jamaludin, T. Kadir, and A. Zisserman, "SpineNet: automatically pinpointing classification evidence in spinal MRIs," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, (Athens, Greece), pp. 166–175, Springer, 2016.

[204] M. Havaei, N. Guizard, N. Chapados, and Y. Bengio, "Hemis: heteromodal image segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, (Athens, Greece), pp. 469–477, Springer, 2016.

[205] M. Havaei, A. Davy, D. Warde-Farley, A. Biard, A. Courville, Y. Bengio, C. Pal, P.-M. Jodoin, and H. Larochelle, "Brain tumor segmentation with deep neural networks," *Medical Image Analysis*, vol. 35, pp. 18–31, 2017.

[206] D. Zikic, Y. Ioannou, M. Brown, and A. Criminisi, "Segmentation of brain tumor tissues with convolutional neural networks," in *MICCAI Workshop on Multimodal Brain Tumor Segmentation Challenge*, (Boston, MA, USA), pp. 36–39, Springer, 2014.

[207] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: convolutional networks for biomedical image segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, (Munich, Germany), pp. 234–241, Springer, 2015.

[208] H. R. Roth, L. Lu, A. Seff, K. M. Cherry, J. Hoffman, S. Wang, J. Liu, E. Turkbey, and R. M. Summers, "A new 2.5D representation for lymph node detection using random sets of deep convolutional neural network observations," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, (Boston, MA, USA), pp. 520–527, Springer, 2014.

[209] H. Chen, Q. Dou, X. Wang, J. Qin, J. C. Cheng, and P.-A. Heng, "3D fully convolutional networks for intervertebral disc localization and segmentation," in *International Conference on Medical Imaging and Augmented Reality*, (Bern, Switzerland), pp. 375–382, Springer, 2016.

[210] K. Kamnitsas, C. Ledig, V. F. Newcombe, J. P. Simpson, A. D. Kane, D. K. Menon, D. Rueckert, and B. Glocker, "Efficient multi-scale 3D CNN with fully connected CRF for accurate brain lesion segmentation," *Medical Image Analysis*, vol. 36, pp. 61–78, 2017.

[211] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, "3D U-Net: learning dense volumetric segmentation from sparse annotation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, (Athens, Greece), pp. 424–432, Springer, 2016.

[212] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-Net: fully convolutional neural networks for volumetric medical image segmentation," in *2016 Fourth International Conference on 3D Vision*, (Stanford, CA, USA), pp. 565–571, IEEE, 2016.

[213] H. Chen, Q. Dou, L. Yu, J. Qin, and P.-A. Heng, "VoxResNet: deep voxel-wise residual networks for brain segmentation from 3D MR images," *NeuroImage*, vol. 170, pp. 446–455, 2018.

[214] K. H. Cha, L. M. Hadjiiski, R. K. Samala, H.-P. Chan, R. H. Cohan, E. M. Caoili, C. Paramagul, A. Alva, and A. Z. Weizer, "Bladder cancer segmentation in CT for treatment response assessment: application of deep-learning convolution neural network–a pilot study," *Tomography*, vol. 2, no. 4, pp. 421–429, 2016.

[215] A. Alansary, K. Kamnitsas, A. Davidson, R. Khlebnikov, M. Rajchl, C. Malamateniou, M. Rutherford, J. V. Hajnal, B. Glocker, D. Rueckert, *et al.*, "Fast fully automatic segmentation of the human placenta from motion corrupted MRI," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, (Athens, Greece), pp. 589–597, Springer, 2016.

[216] M. Shakeri, S. Tsogkas, E. Ferrante, S. Lippe, S. Kadoury, N. Paragios, and I. Kokkinos, "Sub-cortical brain structure segmentation using F-CNN's," in *2016 IEEE 13th International Symposium on Biomedical Imaging*, (Prague, Czech Republic), pp. 269–272, IEEE, 2016.

[217] L. G. Shapiro, "Connected component labeling and adjacency graph construction," in *Machine Intelligence and Pattern Recognition*, vol. 19, pp. 1–30, Elsevier, 1996.

[218] C. Bayan, J. M. Levitt, E. Miller, D. Kaplan, and I. Georgakoudi, "Fully automated, quantitative, noninvasive assessment of collagen fiber content and organization in thick collagen gels," *Journal of Applied Physics*, vol. 105, no. 10, p. 102042, 2009.

[219] L. Zhang and W. Yu, "Orientation image analysis of electrospun submicrofibers based on Hough transform and Regionprops function," *Textile Research Journal*, vol. 87, no. 18, pp. 2263–2274, 2017.

[220] Y. Gowayed, D. V. Schreibman, and M. Roberts, "Surface inspection of textile composite materials using image analysis techniques," *Journal of Composites, Technology and Research*, vol. 18, no. 1, pp. 3–14, 1996.

[221] W. J. Karlon, P.-P. Hsu, S. Li, S. Chien, A. D. McCulloch, and J. H. Omens, "Measurement of orientation and distribution of cellular alignment and cytoskeletal organization," *Annals of Biomedical Engineering*, vol. 27, no. 6, pp. 712–720, 1999.

[222] W. J. Karlon, J. W. Covell, A. D. Mcculloch, J. J. Hunter, and J. H. Omens, "Automated measurement of myofiber disarray in transgenic mice with ventricular expression of ras," *The Anatomical Record: An Official Publication of the American Association of Anatomists*, vol. 252, no. 4, pp. 612–625, 1998.

[223] C. H. Lee, H. J. Shin, I. H. Cho, Y.-M. Kang, I. A. Kim, K.-D. Park, and J.-W. Shin, "Nanofiber alignment and direction of mechanical strain affect the ECM production of human ACL fibroblast," *Biomaterials*, vol. 26, no. 11, pp. 1261–1270, 2005.

[224] 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. 83–92, 2017.

[225] A. D'Amore, J. A. Stella, W. R. Wagner, and M. S. Sacks, "Characterization of the complete fiber network topology of planar fibrous tissues and scaffolds," *Biomaterials*, vol. 31, no. 20, pp. 5345–5354, 2010.

[226] B. Pourdeyhimi, R. Dent, and H. Davis, "Measuring fiber orientation in nonwovens part III: Fourier transform," *Textile Research Journal*, vol. 67, no. 2, pp. 143–151, 1997.

[227] M. Tunák, J. Antoch, J. Kula, and J. Chvojka, "Estimation of fiber system orientation for nonwoven and nanofibrous layers: local approach based on image analysis," *Textile Research Journal*, vol. 84, no. 9, pp. 989–1006, 2014.

[228] C. E. Ayres, B. S. Jha, H. Meredith, J. R. Bowman, G. L. Bowlin, S. C. Henderson, and D. G. Simpson, "Measuring fiber alignment in electrospun scaffolds: a user's guide to the 2D fast Fourier transform approach," *Journal of Biomaterials Science, Polymer Edition*, vol. 19, no. 5, pp. 603–621, 2008.

[229] L. C. Lins, F. Wianny, S. Livi, C. Dehay, J. Duchet-Rumeau, and J.-F. Gérard, "Effect of polyvinylidene fluoride electrospun fiber orientation on neural stem cell differentiation," *Journal of Biomedical Materials Research Part B: Applied Biomaterials*, vol. 105, no. 8, pp. 2376–2393, 2017.

[230] M. Putti, M. Simonet, R. Solberg, and G. W. Peters, "Electrospinning poly (ε-caprolactone) under controlled environmental conditions: influence on fiber morphology and orientation," *Polymer*, vol. 63, pp. 189–195, 2015.

[231] J. A. Paten, G. E. Tilburey, E. A. Molloy, R. Zareian, C. V. Trainor, and J. W. Ruberti, "Utility of an optically-based, micromechanical system for printing collagen fibers," *Biomaterials*, vol. 34, no. 11, pp. 2577–2587, 2013.

[232] P. P. van Zuijlen, H. J. de Vries, E. N. Lamme, J. E. Coppens, J. van Marle, R. W. Kreis, and E. Middelkoop, "Morphometry of dermal collagen orientation by fourier analysis is superior to multi-observer assessment," *The Journal of Pathology: A Journal of the Pathological Society of Great Britain and Ireland*, vol. 198, no. 3, pp. 284–291, 2002.

[233] G. Regnier, D. Dray, E. Jourdain, S. Le Roux, and F. Schmidt, "A simplified method to determine the 3D orientation of an injection molded fiber-filled polymer," *Polymer Engineering & Science*, vol. 48, no. 11, pp. 2159–2168, 2008.

[234] Y. Abdin, S. V. Lomov, A. Jain, G. Van Lenthe, and I. Verpoest, "Geometrical characterization and micro-structural modeling of short steel fiber composites," *Composites Part A: Applied Science and Manufacturing*, vol. 67, pp. 171–180, 2014.

[235] P. Soltani, M. S. Johari, and M. Zarrebini, "3D fiber orientation characterization of nonwoven fabrics using X-ray micro-computed tomography," *World Journal of Textile Engineering and Technology*, vol. 1, pp. 41–47, 2015.

[236] C. Eberhardt and A. Clarke, "Fibre-orientation measurements in short-glass-fibre composites. Part I: automated, high-angular-resolution measurement by confocal microscopy," *Composites Science and Technology*, vol. 61, no. 10, pp. 1389–1400, 2001.

[237] R. Blanc, C. Germain, P. Baylou, M. Cataldi, *et al.*, "Fiber orientation measurements in composite materials," *Composites Part A: applied science and manufacturing*, vol. 37, no. 2, pp. 197–206, 2006.

[238] G. M. Vélez-García, P. Wapperom, D. G. Baird, A. O. Aning, and V. Kunc, "Unambiguous orientation in short fiber composites over small sampling area in a center-gated disk," *Composites Part A: Applied Science and Manufacturing*, vol. 43, no. 1, pp. 104–113, 2012.

[239] H. Altendorf, E. Decencière, D. Jeulin, P. D. S. Peixoto, A. Deniset-Besseau, E. Angelini, G. Mosser, and M.-C. Schanne-Klein, "Imaging and 3D morphological analysis of collagen fibrils," *Journal of Microscopy*, vol. 247, no. 2, pp. 161–175, 2012.

[240] K. P. Quinn, A. Golberg, G. F. Broelsch, S. Khan, M. Villiger, B. Bouma, W. G. Austen Jr, R. L. Sheridan, M. C. Mihm Jr, M. L. Yarmush, *et al.*, "An automated image processing method to quantify collagen fibre organization within cutaneous scar tissue," *Experimental Dermatology*, vol. 24, no. 1, pp. 78–80, 2015.

[241] P. D. Lösel, T. van de Kamp, A. Jayme, A. Ershov, T. Faragó, O. Pichler, N. T. Jerome, N. Aadepu, S. Bremer, S. A. Chilingaryan, *et al.*, "Introducing Biomedisa as an open-source online platform for biomedical image segmentation," *Nature Communications*, vol. 11, no. 1, pp. 1–14, 2020.

[242] F. De Chaumont, S. Dallongeville, N. Chenouard, N. Hervé, S. Pop, T. Provoost, V. Meas-Yedid, P. Pankajakshan, T. Lecomte, Y. Le Montagner, *et al.*, "Icy: an open bioimage informatics platform for extended reproducible research," *Nature Methods*, vol. 9, no. 7, p. 690, 2012.

[243] C. A. Schneider, W. S. Rasband, and K. W. Eliceiri, "NIH Image to ImageJ: 25 years of image analysis," *Nature Methods*, vol. 9, no. 7, pp. 671–675, 2012.

[244] A. E. Carpenter, T. R. Jones, M. R. Lamprecht, C. Clarke, I. H. Kang, O. Friman, D. A. Guertin, J. H. Chang, R. A. Lindquist, J. Moffat, and Others, "CellProfiler: image analysis software for identifying and quantifying cell phenotypes," *Genome Biology*, vol. 7, no. 10, p. R100, 2006.

[245] S. Pieper, M. Halle, and R. Kikinis, "3D Slicer," in *IEEE International Symposium on Biomedical Imaging: Nano to Macro*, (Arlington, VA, USA), pp. 632–635, IEEE, 2004.

[246] C. Sommer, C. Straehle, U. Kothe, and F. A. Hamprecht, "Ilastik: interactive learning and segmentation toolkit," in *IEEE International Symposium on Biomedical Imaging: Nano to Macro*, (Chicago, IL, USA), pp. 230–233, IEEE, 2011.

[247] I. Wolf, M. Vetter, I. Wegner, T. Böttger, M. Nolden, M. Schöbinger, M. Hastenteufel, T. Kunert, and H.-P. Meinzer, "The medical imaging interaction toolkit," *Medical Image Analysis*, vol. 9, no. 6, pp. 594–604, 2005.

[248] M. R. Berthold, N. Cebron, F. Dill, T. R. Gabriel, T. Kötter, T. Meinl, P. Ohl, K. Thiel, and B. Wiswedel, "KNIME–the Konstanz information miner: version 2.0 and beyond," *ACM SIGKDD Explorations Newsletter*, vol. 11, no. 1, pp. 26–31, 2009.

[249] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software: an update," *ACM SIGKDD Explorations Newsletter*, vol. 11, no. 1, pp. 10–18, 2009.

[250] A. Bartschat, E. Hübner, M. Reischl, R. Mikut, and J. Stegmaier, "XPIWIT - an XML pipeline wrapper for the Insight Toolkit," *Bioinformatics*, vol. 32, no. 2, pp. 315–317, 2015.

[251] F. Jug, T. Pietzsch, S. Preibisch, and P. Tomancak, "Bioimage informatics in the context of Drosophila research," *Methods*, vol. 68, no. 1, pp. 60–73, 2014.

[252] R. Mikut, T. Dickmeis, W. Driever, P. Geurts, F. A. Hamprecht, B. X. Kausler, M. J. Ledesma-Carbayo, R. Marée, K. Mikula, P. Pantazis, O. Ronneberger, A. Santos, R. Stotzka, U. Strähle, and N. Peyriéras, "Automated processing of zebrafish imaging data: a survey," *Zebrafish*, vol. 10, no. 3, pp. 401–21, 2013.

[253] M. J. Clarkson, G. Zombori, S. Thompson, J. Totz, Y. Song, M. Espak, S. Johnsen, D. Hawkes, and S. Ourselin, "The NifTK software platform for image-guided interventions: platform overview and NiftyLink messaging," *International Journal of Computer Assisted Radiology and Surgery*, vol. 10, no. 3, pp. 301–316, 2015.

[254] A. Mehrtash, M. Pesteie, J. Hetherington, P. A. Behringer, T. Kapur, W. M. Wells, R. Rohling, A. Fedorov, and P. Abolmaesumi, "DeepInfer: open-source deep learning deployment toolkit for image-guided therapy," in *Medical Imaging 2017: Image-Guided Procedures, Robotic Interventions, and Modeling*, (Houston, TX, USA), p. 101351K, International Society for Optics and Photonics, 2017.

[255] S. Chetlur, C. Woolley, P. Vandermersch, J. Cohen, J. Tran, B. Catanzaro, and E. Shelhamer, "cuDNN: efficient primitives for deep learning," *arXiv preprint arXiv:1410.0759*, 2014.

[256] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, and Others, "Tensorflow: large-scale machine learning on heterogeneous distributed systems," *arXiv preprint arXiv:1603.04467*, 2016.

[257] F. Bastien, P. Lamblin, R. Pascanu, J. Bergstra, I. Goodfellow, A. Bergeron, N. Bouchard, D. Warde-Farley, and Y. Bengio, "Theano: new features and speed improvements," *arXiv preprint arXiv:1211.5590*, 2012.

[258] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: convolutional architecture for fast feature embedding," in *Proceedings of the 22nd ACM International Conference on Multimedia*, (Orlando, FL, USA), pp. 675–678, ACM, 2014.

[259] R. Collobert, K. Kavukcuoglu, and C. Farabet, "Torch7: a matlab-like environment for machine learning," in *Big Learning 2011 : NIPS 2011 Workshop on Algorithms, Systems, and Tools for Learning at Scale*, no. EPFL-CONF-192376, (Sierra Nevada, Spain), NIPS, 2011.

[260] P. Adam, G. Sam, C. Soumith, C. Gregory, Y. Edward, D. Zachary, L. Zeming, D. Alban, A. Luca, and L. Adam, "Automatic differentiation in PyTorch," in *NIPS Autodiff Workshop: The Future of Gradient-based Machine Learning Software and Techniques*, (Long Beach, CA, USA), p. 103, NIPS, 2017.

[261] F. Seide and A. Agarwal, "CNTK: Microsoft's open-source deep-learning toolkit," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, (San Francisco, CA, USA), p. 2135, ACM, 2016.

[262] A. Vedaldi and K. Lenc, "Matconvnet: convolutional neural networks for matlab," in *Proceedings of the 23rd ACM international conference on Multimedia*, (New York, NY, USA), pp. 689–692, ACM, 2015.

[263] F. Chollet *et al.*, "Keras: the python deep learning library," 2018, URL: https://keras.io, Accessed on 15.06.2019.

[264] H. Dong, A. Supratak, L. Mai, F. Liu, A. Oehmichen, S. Yu, and Y. Guo, "TensorLayer: a versatile library for efficient deep learning development," in *Proceedings of the 2017 ACM on Multimedia Conference*, (New York, NY, USA), pp. 1201–1204, ACM, 2017.

[265] S. Dieleman, J. Schlüter, C. Raffel, E. Olson, S. K. Sønderby, D. Nouri, D. Maturana, M. Thoma, E. Battenberg, J. Kelly, and Others, "Lasagne: first release," *Zenodo*, vol. 3, 2015.

[266] E. Gibson, W. Li, C. Sudre, L. Fidon, D. Shakir, G. Wang, Z. Eaton-Rosen, R. Gray, T. Doel, Y. Hu, and Others, "NiftyNet: a deep-learning platform for medical imaging," *arXiv preprint arXiv:1709.03485*, 2017.

[267] N. Pawlowski, S. Ira Ktena, M. C. H. Lee, B. Kainz, D. Rueckert, B. Glocker, and M. Rajchl, "DLTK: state of the art reference implementations for deep learning on medical images," *arXiv preprint arXiv:1711.06853*, 2017.

[268] J. Weissenbock, 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," in *2014 IEEE Pacific Visualization Symposium*, (Yokohama, Japan), pp. 153–160, IEEE, 2014.

[269] N. A. Hotaling, K. Bharti, H. Kriel, and C. G. Simon, "DiameterJ: a validated open source nanofiber diameter measurement tool," *Biomaterials*, vol. 61, pp. 327–338, 2015.

[270] A. Boudaoud, A. Burian, D. Borowska-Wykręt, M. Uyttewaal, R. Wrzalik, D. Kwiatkowska, and O. Hamant, "FibrilTool, an ImageJ plug-in to quantify fibrillar structures in raw microscopy images," *Nature Protocols*, vol. 9, no. 2, p. 457, 2014.

[271] Z. Püspöki, M. Storath, D. Sage, and M. Unser, "Transforms and operators for directional bioimage analysis: a survey," in *Focus on Bio-Image Informatics*, pp. 69–93, Springer, 2016.

[272] O. Ishaq, S. K. Sadanandan, and C. Wählby, "Deep fish: deep learning–based classification of zebrafish deformation for high-throughput screening," *SLAS DISCOVERY: Advancing Life Sciences R&D*, vol. 22, no. 1, pp. 102–107, 2017.

[273] G. Tyagi, N. Patel, and I. Sethi, "A fine-tuned convolution neural network based approach for phenotype classification of zebrafish embryo," *Procedia Computer Science*, vol. 126, pp. 1138–1144, 2018.

[274] N. Jeanray, R. Marée, B. Pruvot, O. Stern, P. Geurts, L. Wehenkel, and M. Muller, "Phenotype classification of zebrafish embryos by supervised learning," *PLoS One*, vol. 10, no. 1, 2015.

[275] B. Al-Saaidah, W. Al-Nuaimy, M. Al-Taee, I. Young, and Q. Al-Jubouri, "Identification of tail curvature malformation in zebrafish embryos," in *2017 8th International Conference on Information Technology (ICIT)*, (Amman, Jordan), pp. 588–593, IEEE, 2017.

[276] R. Alshut, J. Legradi, U. Liebel, L. Yang, J. van Wezel, U. Strähle, R. Mikut, and M. Reischl, "Methods for automated high-throughput toxicity testing using zebrafish embryos," in *Annual Conference on Artificial Intelligence*, (Atlanta, GA, USA), pp. 219–226, Springer, 2010.

[277] A. Singh, B. Ganapathysubramanian, A. K. Singh, and S. Sarkar, "Machine learning for high-throughput stress phenotyping in plants," *Trends in Plant Science*, vol. 21, no. 2, pp. 110–124, 2016.

[278] N. Powell, *Automated morphometric analysis and phenotyping of mouse brains from structural μMR images.* PhD thesis, University College London, 2016.

[279] T. F. Meehan, N. Conte, D. B. West, J. O. Jacobsen, J. Mason, J. Warren, C.-K. Chen, I. Tudose, M. Relac, P. Matthews, *et al.*, "Disease model discovery from 3,328 gene knockouts by the international mouse phenotyping consortium," *Nature Genetics*, vol. 49, no. 8, p. 1231, 2017.

[280] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[281] B. Jahne, *Practical handbook on image processing for scientific and technical applications.* Boca Raton, Florida, USA: CRC Press, 2004.

[282] F. Maes, D. Vandermeulen, and P. Suetens, "Comparative evaluation of multiresolution optimization strategies for multimodality image registration by maximization of mutual information," *Medical Image Analysis*, vol. 3, no. 4, pp. 373–386, 1999.

[283] S. Klein, M. Staring, and J. P. Pluim, "Evaluation of optimization methods for nonrigid medical image registration using mutual information and b-splines," *IEEE Transactions on Image Processing*, vol. 16, no. 12, pp. 2879–2890, 2007.

[284] P. Thevenaz, U. E. Ruttimann, and M. Unser, "A pyramid approach to sub-pixel registration based on intensity," *IEEE Transactions on Image Processing*, vol. 7, no. 1, pp. 27–41, 1998.

[285] C. Studholme, D. L. Hill, and D. J. Hawkes, "Automated 3-D registration of MR and CT images of the head," *Medical Image Analysis*, vol. 1, no. 2, pp. 163–175, 1996.

[286] G. P. Penney, J. Weese, J. A. Little, P. Desmedt, D. L. Hill, *et al.*, "A comparison of similarity measures for use in 2-D-3-D medical image registration," *IEEE Transactions on Medical Imaging*, vol. 17, no. 4, pp. 586–595, 1998.

[287] P. Viola and W. M. Wells III, "Alignment by maximization of mutual information," *International Journal of Computer Vision*, vol. 24, no. 2, pp. 137–154, 1997.

[288] F. Maes, A. Collignon, D. Vandermeulen, G. Marchal, and P. Suetens, "Multimodality image registration by maximization of mutual information," *IEEE Transactions on Medical Imaging*, vol. 16, no. 2, pp. 187–198, 1997.

[289] C. Studholme, D. L. Hill, and D. J. Hawkes, "An overlap invariant entropy measure of 3D medical image alignment," *Pattern Recognition*, vol. 32, no. 1, pp. 71–86, 1999.

[290] J. West, J. M. Fitzpatrick, M. Y. Wang, B. M. Dawant, C. R. Maurer Jr, R. M. Kessler, R. J. Maciunas, C. Barillot, D. Lemoine, A. Collignon, *et al.*, "Comparison and evaluation of retrospective intermodality brain image registration techniques," *Journal of Computer Assisted Tomography*, vol. 21, no. 4, pp. 554–568, 1997.

[291] J. P. Pluim, J. A. Maintz, and M. A. Viergever, "Interpolation artefacts in mutual information-based image registration," *Computer Vision and Image Understanding*, vol. 77, no. 2, pp. 211–232, 2000.

[292] H. Lester and S. R. Arridge, "A survey of hierarchical non-linear medical image registration," *Pattern Recognition*, vol. 32, no. 1, pp. 129–149, 1999.

[293] S. Klein, M. Staring, K. Murphy, M. A. Viergever, and J. P. Pluim, "Elastix: a toolbox for intensity-based medical image registration," *IEEE Transactions on Medical Imaging*, vol. 29, no. 1, pp. 196–205, 2009.

[294] P. Thévenaz and M. Unser, "Optimization of mutual information for multiresolution image registration," *IEEE Transactions on Image Processing*, vol. 9, no. 12, pp. 2083–2099, 2000.

[295] H. Neemuchwala, A. Hero, and P. Carson, "Image matching using alpha-entropy measures and entropic graphs," *Signal Processing*, vol. 85, no. 2, pp. 277–296, 2005.

[296] M. Staring, U. A. Van Der Heide, S. Klein, M. A. Viergever, and J. P. Pluim, "Registration of cervical MRI using multifeature mutual information," *IEEE Transactions on Medical Imaging*, vol. 28, no. 9, pp. 1412–1421, 2009.

[297] A. P. Zijdenbos, B. M. Dawant, R. A. Margolin, and A. C. Palmer, "Morphometric analysis of white matter lesions in MR images: method and validation," *IEEE Transactions on Medical Imaging*, vol. 13, no. 4, pp. 716–724, 1994.

[298] R. Bhagalia, J. A. Fessler, and B. Kim, "Accelerated nonrigid intensity-based image registration using importance sampling," *IEEE Transactions on Medical Imaging*, vol. 28, no. 8, pp. 1208–1216, 2009.

[299] P. Thévenaz, M. Bierlaire, and M. Unser, "Halt on sampling for image registration based on mutual information," *Sampling Theory in Signal and Image Processing*, vol. 7, no. 2, pp. 141–171, 2008.

[300] B. Likar and F. Pernuš, "A hierarchical approach to elastic registration based on mutual information," *Image and Vision Computing*, vol. 19, no. 1-2, pp. 33–44, 2001.

[301] P. Thévenaz, T. Blu, and M. Unser, "Interpolation revisited," *IEEE Transactions on Medical Imaging*, vol. 19, no. 7, pp. 739–758, 2000.

[302] M. Unser, "Splines: a perfect fit for signal processing," in *2000 10th European Signal Processing Conference*, (Tampere, Finland), pp. 22–38, IEEE, 2000.

[303] X. Zhou, T. Hayashi, T. Hara, H. Fujita, R. Yokoyama, T. Kiryu, and H. Hoshi, "Automatic segmentation and recognition of anatomical lung structures from high-resolution chest CT images," *Computerized Medical Imaging and Graphics*, vol. 30, no. 5, pp. 299–313, 2006.

[304] S. Schlüter, A. Sheppard, K. Brown, and D. Wildenschild, "Image processing of multiphase images obtained via X-ray microtomography: a review," *Water Resources Research*, vol. 50, no. 4, pp. 3615–3639, 2014.

[305] M. Walczak, I. Burda, J. Nalepa, and M. Kawulok, "Segmenting lungs from whole-body CT scans," in *International Conference: Beyond Databases, Architectures and Structures*, vol. 716, (Ustroń, Poland), pp. 403–414, Springer, 2017.

[306] C. Heinzl, A. Amirkhanov, and J. Kastner, "Processing, analysis and visualization of CT data," in *Industrial X-Ray Computed Tomography*, pp. 99–142, Springer, 2018.

[307] H. Koyuncu, R. Ceylan, M. Sivri, and H. Erdogan, "An efficient pipeline for abdomen segmentation in CT images," *Journal of Digital Imaging*, vol. 31, no. 2, pp. 262–274, 2018.

[308] A. Buades, B. Coll, and J.-M. Morel, "A non-local algorithm for image denoising," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 2, (San Diego, CA, USA), pp. 60–65, IEEE, 2005.

[309] W. Xu and K. Mueller, "Evaluating popular non-linear image processing filters for their use in regularized iterative CT," in *IEEE Nuclear Science Symposuim & Medical Imaging Conference*, (Knoxville, TN, USA), pp. 2864–2865, IEEE, 2010.

[310] S. Chen and H. Leung, "Chaotic spread spectrum watermarking for remote sensing images," *Journal of Electronic Imaging*, vol. 13, no. 1, pp. 220–231, 2004.

[311] S. U. Lee, S. Y. Chung, and R. H. Park, "A comparative performance study of several global thresholding techniques for segmentation," *Computer Vision, Graphics, and Image Processing*, vol. 52, no. 2, pp. 171–190, 1990.

[312] A. Clare and R. D. King, "Knowledge discovery in multi-label phenotype data," in *Principles of Data Mining and Knowledge Discovery*, pp. 42–53, 2001.

[313] H. R. Roth, L. Lu, A. Farag, A. Sohn, and R. M. Summers, "Spatial aggregation of holistically-nested networks for automated pancreas segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, (Athens, Greece), pp. 451–459, Springer, 2016.

[314] P. Moeskops, J. M. Wolterink, B. H. van der Velden, K. G. Gilhuijs, T. Leiner, M. A. Viergever, and I. Išgum, "Deep learning for multi-task medical image segmentation in multiple modalities," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, (Athens, Greece), pp. 478–486, Springer, 2016.

[315] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (Las Vegas, NV, USA), pp. 770–778, IEEE, 2016.

[316] Y. Song, L. Zhang, S. Chen, D. Ni, B. Lei, and T. Wang, "Accurate segmentation of cervical cytoplasm and nuclei based on multiscale convolutional network and graph partitioning," *IEEE Transactions on Biomedical Engineering*, vol. 62, no. 10, pp. 2421–2433, 2015.

[317] J. Cai, L. Lu, Z. Zhang, F. Xing, L. Yang, and Q. Yin, "Pancreas segmentation in MRI using graph-based decision fusion on convolutional neural networks," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, (Athens, Greece), pp. 442–450, Springer, 2016.

[318] P. F. Christ, M. E. A. Elshaer, F. Ettlinger, S. Tatavarty, M. Bickel, P. Bilic, M. Rempfler, M. Armbruster, F. Hofmann, M. D'Anastasi, *et al.*, "Automatic liver and lesion segmentation in CT using cascaded fully convolutional neural networks and 3D conditional random fields," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, (Athens, Greece), pp. 415–423, Springer, 2016.

[319] Q. Dou, H. Chen, L. Yu, L. Zhao, J. Qin, D. Wang, V. C. Mok, L. Shi, and P.-A. Heng, "Automatic detection of cerebral microbleeds from MR images via 3D convolutional neural networks," *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, pp. 1182–1195, 2016.

[320] H. Fu, Y. Xu, S. Lin, D. W. K. Wong, and J. Liu, "DeepVessel: retinal vessel segmentation via deep learning and conditional random field," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, (Athens, Greece), pp. 132–139, Springer, 2016.

[321] M. Gao, Z. Xu, L. Lu, A. Wu, I. Nogues, R. M. Summers, and D. J. Mollura, "Segmentation label propagation using deep convolutional neural networks and dense conditional random field," in *2016 IEEE 13th International Symposium on Biomedical Imaging (ISBI)*, (Prague, Czech Republic), pp. 1265–1268, IEEE, 2016.

[322] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected CRFs," *arXiv preprint arXiv:1412.7062*, pp. 1–14, 2014.

[323] Q. Dou, H. Chen, Y. Jin, L. Yu, J. Qin, and P.-A. Heng, "3D deeply supervised network for automatic liver segmentation from CT volumes," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, (Athens, Greece), pp. 149–157, Springer, 2016.

[324] T. S. Newman and H. Yi, "A survey of the marching cubes algorithm," *Computers & Graphics*, vol. 30, no. 5, pp. 854–879, 2006.

[325] S. D. ASTM, "Standard test method for flat particles, elongated particles, or flat and elongated particles in coarse aggregate," 2010.

[326] H. Wadell, "Volume, shape, and roundness of rock particles," *The Journal of Geology*, vol. 40, no. 5, pp. 443–451, 1932.

[327] A. Rosenfeld and J. L. Pfaltz, "Sequential operations in digital picture processing," *Journal of the ACM*, vol. 13, no. 4, pp. 471–494, 1966.

[328] B.-K. Jang and R. T. Chin, "Analysis of thinning algorithms using mathematical morphology," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 6, pp. 541–551, 1990.

[329] W. Xie, R. P. Thompson, and R. Perucchio, "A topology-preserving parallel 3D thinning algorithm for extracting the curve skeleton," *Pattern Recognition*, vol. 36, no. 7, pp. 1529–1544, 2003.

[330] R. Shkarin, "Deep learning segmentation framework for volumetric data," Zenodo, 2019, URL: https://zenodo.org/record/3635010, Accessed on 03.02.2020.

[331] B. B. Avants, N. Tustison, and G. Song, "Advanced normalization tools (ANTS)," *Insight Journal*, vol. 2, no. 365, pp. 1–35, 2009.

[332] R. Shkarin, "X-Ray CT phantom dataset," Zenodo, 2019, URL: https://zenodo.org/record/3594972, Accessed on 29.12.2019.

[333] D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[334] N. E. Persson, M. A. McBride, M. A. Grover, and E. Reichmanis, "Automated analysis of orientational order in images of fibrillar materials," *Chemistry of Materials*, vol. 29, no. 1, pp. 3–14, 2017.

[335] J. Kastner and C. Heinzl, "X-ray computed tomography for non-destructive testing and materials characterization," in *Integrated Imaging and Vision Techniques for Industrial Inspection*, pp. 227–250, Springer, 2015.

[336] C. Pudney, "Distance-ordered homotopic thinning: a skeletonization algorithm for 3D digital images," *Computer Vision and Image Understanding*, vol. 72, no. 3, pp. 404–413, 1998.

[337] T.-C. Lee, R. L. Kashyap, and C.-N. 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.

[338] K. Palágyi and A. Kuba, "A 3D 6-subiteration thinning algorithm for extracting medial lines," *Pattern Recognition Letters*, vol. 19, no. 7, pp. 613–627, 1998.

[339] B.-J. Zandt, A. Losnegård, E. Hodneland, M. L. Veruki, A. Lundervold, and E. Hartveit, "Semi-automatic 3D morphological reconstruction of neurons with densely branching morphology: application to retinal aii amacrine cells imaged with multi-photon excitation microscopy," *Journal of Neuroscience Methods*, vol. 279, pp. 101–118, 2017.

[340] R. Shkarin, S. Shkarina, V. Weinhardt, R. A. Surmenev, M. A. Surmeneva, A. Shkarin, T. Baumbach, and R. Mikut, "GPU-accelerated ray-casting for 3D fiber orientation analysis," *PLoS One*, vol. 15, no. 7, 2020.

[341] R. Shkarin, A. Shkarin, S. Shkarina, A. Cecilia, R. A. Surmenev, M. A. Surmeneva, V. Weinhardt, T. Baumbach, and R. Mikut, "Quanfima: An open source python package for automated fiber analysis of biomaterials," *PLoS One*, vol. 14, no. 4, 2019.

[342] S. v. d. Walt, S. C. Colbert, and G. Varoquaux, "The NumPy array: a structure for efficient numerical computation," *Computing in Science & Engineering*, vol. 13, no. 2, pp. 22–30, 2011.

[343] E. Jones, T. Oliphant, and P. Peterson, "SciPy: open source scientific tools for python," *Computing in Science and Engineering*, vol. 9, pp. 10–20, 2007.

[344] S. Van der Walt, J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart, and T. Yu, "scikit-image: image processing in python," *PeerJ*, vol. 2, p. e453, 2014.

[345] S. K. Lam, A. Pitrou, and S. Seibert, "Numba," in *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC - LLVM '15*, (Atlanta, GA, USA), pp. 1–6, ACM, 2015.

[346] A. Klöckner, N. Pinto, Y. Lee, B. Catanzaro, P. Ivanov, and A. Fasih, "PyCUDA and PyOpenCL: A scripting-based approach to GPU run-time code generation," *Parallel Computing*, vol. 38, no. 3, pp. 157–174, 2012.

[347] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: A system for large-scale machine learning," in *Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation*, (Savannah, GA, USA), pp. 265–283, PeerJ, 2016.

[348] Wes McKinney, "Data structures for statistical computing in Python," in *Proceedings of the 9th Python in Science Conference*, vol. 445, (Austin, TX, USA), pp. 56 – 61, SciPy USA, 2010.

[349] S. Seabold and J. Perktold, "Statsmodels: econometric and statistical modeling with python," in *Proceedings of the 9th Python in Science Conference*, vol. 57, (Austin, TX, USA), p. 61, SciPy USA, 2010.

[350] B. Josso, D. R. Burton, and M. J. Lalor, "Texture orientation and anisotropy calculation by Fourier transform and principal component analysis," *Mechanical Systems and Signal Processing*, vol. 19, no. 5, pp. 1152–1161, 2005.

[351] H.-C. Tseng, R.-Y. Chang, and C.-H. Hsu, "Phenomenological improvements to predictive models of fiber orientation in concentrated suspensions," *Journal of Rheology*, vol. 57, no. 6, pp. 1597–1631, 2013.

[352] A. Miettinen, A. Ojala, L. Wikström, R. Joffe, B. Madsen, K. Nättinen, and M. Kataja, "Non-destructive automatic determination of aspect ratio and cross-sectional properties of fibres," *Composites Part A: Applied Science and Manufacturing*, vol. 77, pp. 188–194, 2015.

[353] Y. Lee, S. Lee, J. Youn, K. Chung, and T. Kang, "Characterization of fiber orientation in short fiber reinforced composites with an image processing technique," *Materials Research Innovations*, vol. 6, no. 2, pp. 65–72, 2002.

[354] J. D. Hunter, "Matplotlib: a 2D graphics environment," *Computing in Science and Engineering*, vol. 9, no. 3, pp. 90–95, 2007.

[355] P. Ramachandran and G. Varoquaux, "Mayavi: 3D visualization of scientific data," *Computing in Science & Engineering*, vol. 13, no. 2, pp. 40–51, 2011.

[356] N. Zhu, G. Wang, G. Yang, and W. Dai, "A fast 2D Otsu thresholding algorithm based on improved histogram," in *2009 Chinese Conference on Pattern Recognition*, (Nanjing, China), pp. 1–5, IEEE, 2009.

[357] R. Stotzka, V. Hartmann, T. Jejkal, M. Sutter, J. van Wezel, M. Hardt, A. Garcia, R. Kupsch, and S. Bourov, "Perspective of the Large Scale Data Facility (LSDF) supporting nuclear fusion applications," in *2011 19th International Euromicro Conference on Parallel, Distributed and Network-Based Processing*, (Ayia Napa, Cyprus), pp. 373–379, IEEE, 2011.

[358] S. Kirchmaier, K. Naruse, J. Wittbrodt, and F. Loosli, "The genomic and genetic toolbox of the teleost medaka (Oryzias latipes)," *Genetics*, vol. 199, no. 4, pp. 905–918, 2015.

[359] G. J. Lieschke and P. D. Currie, "Animal models of human disease: zebrafish swim into view," *Nature Reviews Genetics*, vol. 8, no. 5, p. 353, 2007.

[360] E. E. Patton and L. I. Zon, "The art and design of genetic screens: zebrafish," *Nature Reviews Genetics*, vol. 2, no. 12, pp. 956–966, 2001.

[361] A. M. Maga, N. Navarro, M. L. Cunningham, and T. C. Cox, "Quantitative trait loci affecting the 3D skull shape and size in mouse and prioritization of candidate genes in-silico," *Frontiers in Physiology*, vol. 6, p. 92, 2015.

[362] M. E. Dickinson, A. M. Flenniken, X. Ji, L. Teboul, M. D. Wong, J. K. White, T. F. Meehan, W. J. Weninger, H. Westerberg, H. Adissu, *et al.*, "High-throughput discovery of novel developmental phenotypes," *Nature*, vol. 537, no. 7621, pp. 508–514, 2016.

[363] A. Velroyen, A. Yaroshenko, D. Hahn, A. Fehringer, A. Tapfer, M. Müller, P. Noël, B. Pauwels, A. Sasov, A. Yildirim, *et al.*, "Grating-based X-ray dark-field computed tomography of living mice," *EBioMedicine*, vol. 2, no. 10, pp. 1500–1506, 2015.

[364] J. Moosmann, A. Ershov, V. Altapova, T. Baumbach, M. S. Prasad, C. LaBonne, X. Xiao, J. Kashef, and R. Hofmann, "X-ray phase-contrast in vivo microtomography probes new aspects of Xenopus gastrulation," *Nature*, vol. 497, no. 7449, pp. 374–377, 2013.

[365] P. Cloetens, W. Ludwig, D. Van Dyck, J.-P. Guigay, M. Schlenker, and J. Baruchel, "Quantitative phase tomography by holographic reconstruction," in *Developments in X-Ray Tomography II*, vol. 3772, (Denver, CO, USA), pp. 279–290, International Society for Optics and Photonics, 1999.

[366] C. David, B. Nöhammer, H. Solak, and E. Ziegler, "Differential X-ray phase contrast imaging using a shearing interferometer," *Applied Physics Letters*, vol. 81, no. 17, pp. 3287–3289, 2002.

[367] M. Brinkmann, L. Y. Rizzo, T. Lammers, F. Gremse, S. Schiwy, F. Kiessling, and H. Hollert, "Micro-computed tomography (μCT) as a novel method in ecotoxicology-determination of morphometric and somatic data in rainbow trout (Oncorhynchus mykiss)," *Science of the Total Environment*, vol. 543, pp. 135–139, 2016.

[368] W. Vågberg, D. H. Larsson, M. Li, A. Arner, and H. M. Hertz, "X-ray phase-contrast tomography for high-spatial-resolution zebrafish muscle imaging," *Scientific Reports*, vol. 5, p. 16625, 2015.

[369] F. Babaei, T. L. C. Hong, K. Yeung, S. H. Cheng, and Y. W. Lam, "Contrast-enhanced X-ray micro-computed tomography as a versatile method for anatomical studies of adult zebrafish," *Zebrafish*, vol. 13, no. 4, pp. 310–316, 2016.

[370] R. Shahzad, D. Bos, R. P. Budde, K. Pellikaan, W. J. Niessen, A. van der Lugt, and T. van Walsum, "Automatic segmentation and quantification of the cardiac structures from non-contrast-enhanced cardiac CT scans," *Physics in Medicine & Biology*, vol. 62, no. 9, p. 3798, 2017.

[371] Y. Xin, G. Song, M. Cereda, S. Kadlecek, H. Hamedani, Y. Jiang, J. Rajaei, J. Clapp, H. Profka, N. Meeder, *et al.*, "Semiautomatic segmentation of longitudinal computed tomography images in a rat model of lung injury by surfactant depletion," *Journal of Applied Physiology*, vol. 118, no. 3, pp. 377–385, 2015.

[372] D. Yan, Z. Zhang, Q. Luo, and X. Yang, "A novel mouse segmentation method based on dynamic contrast enhanced micro-CT images," *PLoS One*, vol. 12, no. 1, 2017.

[373] V. Weinhardt, R. Shkarin, T. Wernet, J. Wittbrodt, T. Baumbach, and F. Loosli, "Quantitative morphometric analysis of adult teleost fish by X-ray computed tomography," *Scientific Reports*, vol. 8, no. 1, pp. 1–12, 2018.

[374] A. Rack, T. Weitkamp, S. B. Trabelsi, P. Modregger, A. Cecilia, T. dos Santos Rolo, T. Rack, D. Haas, R. Simon, R. Heldele, *et al.*, "The micro-imaging station of the TopoTomo beamline at the ANKA synchrotron light source," *Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms*, vol. 267, no. 11, pp. 1978–1988, 2009.

[375] Y. Cheng, V. Altapova, L. Helfen, F. Xu, T. dos Santos Rolo, P. Vagovič, M. Fiederle, and T. Baumbach, "Multi-contrast computed laminography at ANKA light source," in *Journal of Physics: Conference Series*, vol. 463, (Shanghai, China), p. 012038, IOP Publishing, 2013.

[376] M. Vogelgesang, S. Chilingaryan, T. dos Santos Rolo, and A. Kopmann, "UFO: a scalable GPU-based image processing framework for on-line monitoring," in *2012 IEEE 14th International Conference on High Performance Computing and Communication & 2012 IEEE 9th International Conference on Embedded Software and Systems*, (NW Washington, DC, USA), pp. 824–829, IEEE, 2012.

[377] P. Lösel and V. Heuveline, "Enhancing a diffusion algorithm for 4D image segmentation using local information," in *Medical Imaging 2016: Image Processing*, vol. 9784, (San Diego, CA, USA), p. 97842L, International Society for Optics and Photonics, 2016.

[378] A. Sombke, E. Lipke, P. Michalik, G. Uhl, and S. Harzsch, "Potential and limitations of X-ray micro-computed tomography in arthropod neuroanatomy: a methodological and comparative survey," *Journal of Comparative Neurology*, vol. 523, no. 8, pp. 1281–1295, 2015.

[379] J. Gong, J. Arnold, and S. Cohn, "Composition of trabecular and cortical bone," *The Anatomical Record*, vol. 149, no. 3, pp. 325–331, 1964.

[380] G. Friedlaender, "Bone grafts. The basic science rationale for clinical applications.," *Journal of Bone and Joint Surgery*, vol. 69, no. 5, pp. 786–790, 1987.

[381] B. Dhandayuthapani, Y. Yoshida, T. Maekawa, and D. S. Kumar, "Polymeric scaffolds in tissue engineering application: a review," *International Journal of Polymer Science*, vol. 2011, 2011.

[382] S. Shkarina, R. Shkarin, V. Weinhardt, E. Melnik, G. Vacun, P. J. Kluger, K. Loza, M. Epple, S. I. Ivlev, T. Baumbach, *et al.*, "3D biodegradable scaffolds of polycaprolactone with silicate-containing hydroxyapatite microparticles for bone tissue engineering: high-resolution tomography and in vitro study," *Scientific Reports*, vol. 8, no. 1, pp. 1–13, 2018.

[383] S. N. Gorodzha, M. A. Surmeneva, R. A. Surmenev, M. Gribennikov, V. F. Pichugin, A. A. Sharonova, A. Pustovalova, O. Prymack, M. Epple, A. Wittmar, *et al.*, "Wettability of thin silicate-containing hydroxyapatite films formed by RF-magnetron sputtering," *Russian Physics Journal*, vol. 56, no. 10, pp. 1163–1169, 2014.

[384] M. S. Kim and G. Kim, "Three-dimensional electrospun polycaprolactone (PCL)/alginate hybrid composite scaffolds," *Carbohydrate Polymers*, vol. 114, pp. 213–221, 2014.

[385] A. Sharma, M. S. Molla, K. S. Katti, and D. R. Katti, "Multiscale models of degradation and healing of bone tissue engineering nanocomposite scaffolds," *Journal of Nanomechanics and Micromechanics*, vol. 7, no. 4, p. 04017015, 2017.

[386] Y. S. Cho, M. W. Hong, H.-J. Jeong, S.-J. Lee, Y. Y. Kim, and Y.-S. Cho, "The fabrication of well-interconnected polycaprolactone/hydroxyapatite composite scaffolds, enhancing the exposure of hydroxyapatite using the wire-network molding technique," *Journal of Biomedical Materials Research Part B: Applied Biomaterials*, vol. 105, no. 8, pp. 2315–2325, 2017.

[387] I. Gibson, S. Best, and W. Bonfield, "Chemical characterization of silicon-substituted hydroxyapatite," *Journal of Biomedical Materials Research*, vol. 44, no. 4, pp. 422–428, 1999.

[388] A. M. Pietak, J. W. Reid, M. J. Stott, and M. Sayer, "Silicon substitution in the calcium phosphate bioceramics," *Biomaterials*, vol. 28, no. 28, pp. 4023–4032, 2007.

[389] M. Schindler, I. Ahmed, J. Kamal, A. Nur-E-Kamal, T. H. Grafe, H. Y. Chung, and S. Meiners, "A synthetic nanofibrillar matrix promotes in vivo-like organization and morphogenesis for cells in culture," *Biomaterials*, vol. 26, no. 28, pp. 5624–5631, 2005.

[390] H. A. Metwally, R. V. Ardazishvili, A. N. Severyukhina, A. M. Zaharevich, A. A. Skaptsov, S. B. Venig, G. B. Sukhorukov, and D. A. Gorin, "The influence of hydroxyapatite and calcium carbonate microparticles on the mechanical properties of nonwoven composite materials based on polycaprolactone," *BioNanoScience*, vol. 5, no. 1, pp. 22–30, 2015.

[391] A. Thorvaldsson, H. Stenhamre, P. Gatenholm, and P. Walkenström, "Electrospinning of highly porous scaffolds for cartilage regeneration," *Biomacromolecules*, vol. 9, no. 3, pp. 1044–1049, 2008.

[392] P. Di Tommaso, M. Chatzou, E. W. Floden, P. P. Barja, E. Palumbo, and C. Notredame, "Nextflow enables reproducible computational workflows," *Nature Biotechnology*, vol. 35, no. 4, pp. 316–319, 2017.

[393] T. van de Kamp, P. Vagovič, T. Baumbach, and A. Riedel, "A biological screw in a beetle's leg," *Science*, vol. 333, no. 6038, pp. 52–52, 2011.

[394] J. Stegmaier, *New methods to improve large-scale microscopy image analysis with prior knowledge and uncertainty*. Karlsruhe, Germany: KIT Scientific Publishing, 2017.