Fakultät für Maschinenwesen
Technische Universität München

Concepts for X-ray Based 3D Replication in Rapid Prototyping Environment

Jelena Prša

Vollständiger Abdruck der von der Fakultät für Maschinenwesen der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs (Dr.-Ing.)

genehmigten Dissertation.

Vorsitzender:
Prof. Dr.-Ing. Michael Friedrich Zäh

Prüfende der Dissertation:
1. Prof. Dr. rer. nat. Tim Christian Lüth
2. Prof. Nassir Navab, Ph.D.
3. Prof. Dr.-Ing. Darius Burschka

Die Dissertation wurde am 25.10.2017 bei der Technischen Universität München eingereicht und durch die Fakultät für Maschinenwesen am 07.05.2018 angenommen.

Acknowledgements

This thesis has been developed during my research work at the chair of Micro Technology and Medical Device Technology under the guidance of Prof. Dr. Tim C. Lüth. I would like to thank him for giving me the opportunity to conduct my doctoral thesis at his chair and for supervising me. I am grateful that I had the chance to discover myself within the broad spectrum of the tasks that were assigned to me.

I am thankful to Prof. Dr. Nassir Navab for accepting the position of the second assessor of the doctoral examination and to Prof. Dr. Michael F. Zäh for overtaking the examination chairmanship.

My warm thanks go to Dr. Irlinger, with whom I had many fruitful discussions, both professionally and personally. His pleasant personality, useful advice and encouragement were of great help to me, especially in difficult moments.

I would also like to thank my chair colleagues and to friends Juan, Mo and Joachim for pleasant work breaks. Sharing stories, cakes, tours and mensa lunches with you was a great joy.

Many thanks to my sister Ivana for thorough language proofreading and elimination of typos. Working with her was very uncomplicated, and she had a lot of understanding for time constraints.

My sincerest gratitude goes to my beloved husband Aaron for supporting me in so many ways. Thank you for taking so much time and effort while proofreading and improving my thesis. All our scientific discussions had a great impact on my work. Without them, this thesis would not be the same. Furthermore, your technical help was very appreciated. Last, but not the least, thank you for having so much understanding and for being a great source of motivation. Your support in harsh times meant a world to me.

I am very grateful for having a life of comfort with a lot of possibilities. I want to thank my parents for enabling me that. They, together with my sisters, have always had an ear for my concerns, encouraged me and believed in me. Hvala vam.

Abstract

A fully automated 3D replication of complete objects in Rapid Prototyping (RP) environment currently does not exist. This thesis couples three main computational processes involved in RP 3D Replication: X-ray based digitization, post-processing and Additive Manufacturing (AM). The thesis focuses on the development of RP-compatible procedures for the reconstruction and model preparation for the subsequent selectable AM process.

Regarding the X-ray based digitization, a parallelisable cone-beam reconstruction unit with an integrated X-ray simulator has been developed. An automated setting of the X-ray tube parameters enhanced the procedures. Recognition and correction of planes, direct slicing of voxel-based models, and mesh simplification were developed as a part of the post-processing unit. Concerning AM, the reduction of voids within 3D printed parts has been realised.

The realisation of the proposed concepts leads to accurate 3D replicas and makes the complete replication process fast and user-friendly, but still modular. With the automated computational pipeline for replicating, manufacturing of first 3D replicating devices is feasible.

Zusammenfassung

Derzeit gibt es noch kein Rapid Prototyping (RP)-Verfahren zur vollautomatischen, ganzheitlichen Replikation von 3D-Objekten. Diese Arbeit verzahnt die drei Hauptberechnungsprozesse der RP-3D-Replikation: die Röntgen-basierte Digitalisierung, die Nachberarbeitung und die additive Fertigung. Im Fokus stehen die Entwicklung von RP-kompatiblen Verfahren für die Rekonstruktion und die Vorbereitung für das auswählbare additive Fertigungsverfahren.

Bezüglich der Röntgen-basierten Digitalisierung wurde eine parallelisierbare Rekonstruktionseinheit mit integriertem Röntgen-Simulator entwickelt. Dabei werden die Parameter der Röntgenröhre automatisch eingestellt. Weitere Verfahren wie eine ebenengestützte Oberflächen-Korrektur, ein direktes Slicing der voxelbasierten Modellen und eine Verfeinfachungsmethode der Polygonnetze wurden als Teile der Nachbearbeitung entwickelt. Die tropfenbasierte additive Fertigung wurde durch ein Lückenreduzierungsverfahren verbessert.

Das vorgestellte Konzept realisiert präzise Replikationen und beschleunigt und vereinfacht den Replikationsvorgang ohne Modularitätsverlust. Diese automatisierte Replikations-Pipeline ermöglicht die Herstellung von ersten 3D-Replikationsvorrichtungen.

Contents

1 Introduction

Conventional 2D devices for printing, scanning and copying are well known and used on a regular basis in everyday life. From a digitalized text or graphics 2D printers generate a sheet of paper with the corresponding input information. 2D scanners reverse the order and extract a digitalized information from a sheet of paper. A 2D copy device combines the two processes (scanning followed by printing) and enables copying of the information from one sheet of paper to another one. Since we reside within a 3D world, why not copying 3D objects? This thesis addresses various aspects of the 3D replication process.

1.1 Application

The remarkable development of Rapid Prototyping (RP) in the past decades has opened a door for the fast fabrication of 3D replicas. Rapid prototyping is an umbrella term for various methods for the fast fabrication of prototype components from design data. It links together processes within rapid tooling, Additive Manufacturing (AM), also known as 3D Printing, and reverse engineering.

In conventional engineering, the main goal is to obtain a real part (further referred to as *3D object*) from a model or a general concept. In reverse engineering, however, 3D objects are transformed into 3D (digitized) models. Though reverse engineering as a term covers broad aspects of retrieving information out of existing objects or past events, its restriction to reverse engineering of shape/geometry will be set in focus here.

The main steps of reverse engineering are data acquisition, reconstruction and post-processing. The conventional engineering includes model preparation, fabrication and part finishing. Starting with a 3D object and performing all these steps consecutively would lead to obtaining its replica. Various application fields of the replication procedure include medical device technology (manufacturing of dental prosthesis based on 3D scans of teeth or the teeth concavities), orthopaedics (bio-printing of artificial bones with adjoined stem cells based on a 3D Scan of an individual bone), architecture and arts (digitizing of existing parts serves as a restoring platform), [Fastermann, 2014]. In general, 3D Replication can be employed for any kind of repairing of damaged parts, manufacturing of individualised parts and redesign, [Hagl, 2014].

To fabricate a 3D object, a corresponding 3D (digitized) model is required. Possible data paths for a model creation include 3D CAD (Computer-Aided Design) manual construction, semi-manual 3D CAD construction based on 2D drawings or technical data, and data measurement of actual geometrical objects, [Hagl, 2014]. For the majority of industrial applications, 3D models are obtained via CAD construction, which is time-consuming and demanding due to the potential complex geometries, [Gebhardt and

Hötter, 2016]. A possible workaround is introduced by automatised methods of 3D Digitization (also known as 3D Scanning). Retrieving the shape out of 3D objects has become an important part in wide range of both industrial and research areas. Some examples are medical engineering, aerospace engineering, automotive engineering, architecture and arts, [Wang, 2010].

After obtaining a 3D digitized model, its fabrication can be carried out. AM enables fast and straightforward layer-wise manufacturing of arbitrary 3D objects. Due to its versatility and the possibility to create prototypes and functional parts, AM has rapidly spread and found its application in a large number of fields. They include robotics, aerospace technology, automotive engineering, naval architecture, medical device technology, arts and jewellery design, architecture, furniture industry and textile industry, [Bibb et al., 2010], [Chimento et al., 2011], [Kouhi et al., 2008], [Fastermann, 2014], [Hagl, 2014].

1.2 Motivation

Up to this date, a 3D replication of internal structures in RP environment has not been automatised. There is no single entity that unites the processes for obtaining a complete 3D model out of an existing object and its recreation via AM.

The most of 3D digitization techniques are either optical [Skala et al., 2013], [Voicu et al., 2013] or require physical contact. Therefore they are limited to the determination of the outer (external) geometry of an object. However, 3D digitization techniques based on X-rays enable the examination of the internal structures without contacting probes.

A realisation of a 3D replicating entity which combines X-ray based digitizing with AM does not exist. Not only is the realisation of a 3D replicating entity missing, but the conceptual aspects of adjusting the process of X-ray based reverse engineering to different AM fabrication requirements are missing too. This implies a certain disconnection between model-obtaining and object-creating entities, such as:

Missing precision consensus. A reconstructed 3D model is possibly of much higher precision than the manufacturing precision.

Missing model preparation for manufacturing. No automatic correction of regular geometries nor direct slicing.

Lack of user-friendliness. Many manual steps are unsuitable for RP environment.

This thesis refers to the computational coupling of three main processes involved in RP 3D Replication:

X-ray based digitization — data acquisition and 3D reconstruction.

Post-processing — transformation and correction of the reconstructed data.

AM — modular layerwise fabrication.

The focus is set on the development of RP-adopted procedures for model reconstruction based on X-ray projections and model preparation for subsequent selectable AM processes. Necessary computational aspects concerning the coupling will be subsequently described in more detail.

1.3 Overview of chapters

This thesis consists of the following chapters:

State of the art presents the current state of the processes involved in the replication of 3D objects in a RP environment. Covered topics are: data acquisition, reconstruction algorithms for X-ray based digitizing methods, post-processing (data conversion, simplification, segmentation and fitting), process planning in AM, and available 3D replication tools. The limitations of the current state of the art are stated at the end of the chapter.

X-ray based 3D Replication in RP environment gives a clipped overview of the proposed solutions for the limitations. It focuses on user-controlled reconstruction unit and automated processing of 3D reconstructed models, while regarding the subsequent fabrication process.

Concept introduces detailed structures and procedures of the proposed solutions from the mathematical point of view. New algorithms are introduced: plane correction, direct slicing of voxel-based models, mesh simplification and detection and filling of voids.

Realisation describes the implementation aspects of the introduced algorithms. This chapter focuses on: reconstruction and post-processing unit, X-ray database and reduction of voids.

Experiments and Analysis starts with the demonstration of the obtained replication results, followed by the experimental examination of the influence of different reconstruction parameters. Mechanical evaluation of enhanced AM parts closes the chapter.

2 State of the art

This chapter presents the state of the art of different processes involved in the replication of 3D objects in a RP environment. State of the art of data acquisition and corresponding reconstruction aspects (further referred to as 3D digitizing or 3D digitization) is described in Section 2.1. An overview of different reconstruction algorithms for X-ray based digitizing methods is presented in Section 2.2, followed by typical processes that occur at cone-beam X-ray apparatus in Section 2.3. State of the art of the post-processing aspects, such as data conversion, simplification, segmentation and fitting is described in Section 2.4. In Section 2.5 follows a short overview of AM processes, as well as technology differentiation and aspects of process planing. State of the research and available tools for 3D replication are presented in Section 2.6. The limitations of the aforementioned aspects are finally stated in Section 2.7.

2.1 Methods for digitization of 3D objects

2.1.1 Classification

Methods for 3D digitizing have been classified by various authors, such as [Kumar et al., 2013], [Gockel, 2006] and [Raja and Fernandes, 2007]. An overview of adopted and renewed classification is presented in Figure 2.1.

3D digitizing can be classified into contact and non-contact methods, according to the physical collision between sensors and objects. Contact methods rely on probes which reconstruct the geometry of 3D objects by physical touching and can be either destructive (the objects are damaged after the process) or non-destructive. Non-contact methods are divided according to how the sensing medium interacts with the object into transmissive and reflective methods. In transmissive methods either magnetic fields and radio waves, or X-rays are passing through the medium. In reflective methods the means are either ultrasound or light. Optical methods can be classified as either passive or active, depending on the reflection measurement of either existing ambient light or additional artificial light.

2.1.2 Contact methods

Non-destructive

Methods where a 3D model is obtained by a non-damaging direct contact between a probe and an object are called non-destructive methods. In non-destructive methods a sensing-probe is positioned on three mutually orthogonal movable axes. As the sensing-probe moves along the surface of a 3D object, the distance between each axis relative

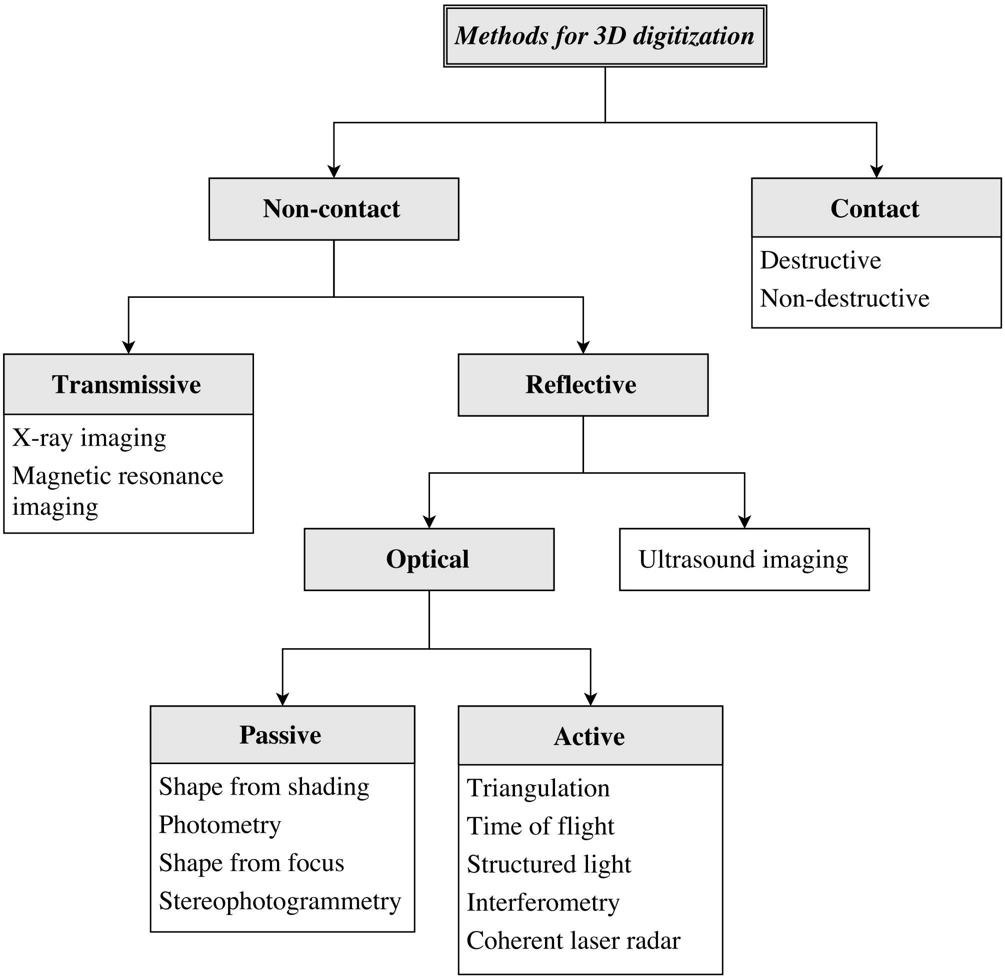


Figure 2.1: Classification of methods for 3D digitization.

to the reference values is measured. These distances are used to obtain points in a 3D space, which define the outer surface of the object geometry. A mechanical realisation of this most commonly used 3D digitizing approach is the coordinate measuring machine (CMM), which has been used since the 1960s. Its main advantages are high precision (up to few microns) and a wide range of applicable materials, [Bidanda and Geng, 2016], [Raja and Fernandes, 2007].

Destructive

Methods where a 3D model is obtained by damaging direct contact between a probe and an object are called destructive methods. In this approach a computer numerical control (CNC) milling machine iteratively subtracts the layers of 3D objects. The subtracted layers are captured by a camera and analysed subsequently. The captured images are afterwards converted to the edge points of the geometry. This technology is also known as cross-sectional scanning. Advantages of this method are fast data acquisition, possibility to capture internal structures and a wide range of applicable materials. Achievable accuracy is in sub-millimetre range, [Raja and Fernandes, 2007].

2.1.3 Transmissive methods

X-ray imaging

The first employments of X-ray (volumetric) imaging were established in the 1970s for medical purposes. As a medical imaging tool it enabled distinguishing between tissue types based on their different densities, [Dössel and Buzug, 2014]. This approach has been extended to material inspection in closed housing and since the 1980s the first X-ray imaging tools for non-medical applications have been available. Nowadays they represent a standard tool in various metrological and manufacturing applications, as well as in material science, [Müller et al., 2013].

In order to digitize a 3D object via X-rays, an X-ray source and a detector are required. A 3D object is typically fixed on a rotational plate between the source and a 2D detector. Once the X-ray source starts emitting the X-rays, the object absorbs the X-rays depending on its geometry, thickness and material. The detector measures the remaining radiation and translates it into 2D grey-scaled images (also called projections). For each discrete rotational angle of a 360°-span, a 2D image of the rotated object is made. Upon obtaining all the necessary 2D images, a 3D reconstruction can be calculated. Details about different reconstruction algorithms are presented in Section 2.2.4. Since X-rays are being transmitted through an object, the inspection is enabled not only of the outer, but also of the inner structures (such as bores, cavities and hollow gaps). Additional advantages are a very high resolution of the measured objects (in sub-micron range) and a high precision (up to few microns), [du Plessis et al., 2016], [Sun et al., 2012].

Magnetic resonance imaging

Magnetic resonance imaging (MRI), like X-ray imaging, has its origins and main application in medical imaging. It has been developed in the 1980s based on the principles of nuclear magnetic resonance. When particular atoms are placed within a magnetic field, they absorb and emit radio frequency energy. While hydrogen atoms in water are most frequently used in medical applications, phosphorus atoms have been applied for 3D digitizing of non-living objects, [Frey et al., 2012]. In order to target phosphorus atoms, a more complicated sequence of radio waves is applied. With that, harder solids with low water content can also be examined. Aside medical applications, this transmissive 3D digitizing method is mostly used in material science. Its main advantages are the ability of capturing internal structures without ionizing radiation and obtaining a very good contrast between different materials, [Jutras and De Zanche, 2016].

2.1.4 Reflective non-optical methods

Ultrasound imaging

Other than X-rays and magnetic resonated atoms, ultrasound waves can be used for 3D digitizing as well. The range of ultrasound wave frequencies for this purposes is typically between 2 MHz and 10 MHz. The first applications of ultrasound waves took place in the 1910s and since then they have expanded in various fields, like sonar, non-destructive testing, material science and biomedical imaging, [Harvey et al., 2014]. The main advantage of ultrasound imaging technique is its fast (raw) data acquisition, [Gholizadeh, 2016], [Elaraj, 2017].

Ultrasound based-devices emit impulses, which are in turn being reflected from the targeted entity. The distance between the ultrasound source and the targeted entity is calculated based on the known sound velocity and time difference between the emitting and receiving moment. Even though ultrasound imaging is not a transmissive technique, it enables the inspection of the internal structures. At the transition area between different materials ultrasound waves will only be partly reflected. The non-reflected portion of the waves continues passing through until it reaches the next material transition. This process can repeat multiple times. The reflected portion is determined by wave impedances of different materials. With this, material borders can be determined and classified.

The invention of 3D ultrasound technique enabled emission of ultrasound waves at different angles, [Von Ramm and Smith, 1987]. Captured 2D images lead to reconstruction of whole 3D models.

2.1.5 Reflective optical methods

Already since the 1860s there have been approaches to reconstruct physical 3D shape of an object by optical means, [Beraldin et al., 2000]. Optical methods are nowadays the most common 3D digitization approach. They rely on the interaction between light and targeted objects.

Active methods use an additional light source, which actively emits light toward the object and the reflected light is measured in order to obtain the outer geometry. Most systems that employ active optical approach for the 3D digitization have performance in the sub-millimetre range of accuracy, [Raja and Fernandes, 2007]. In this group of methods belong *triangulation*, *time of flight*, *structured light*, *interferometry* and *coherent laser radar*.

Passive optical methods, on the contrary, rely on the reflected ambient light. The depth of an object can be retrieved either by capturing multiple images with a monocular camera under varied conditions, or with a binocular stereo camera, [Pertuz et al., 2013]. Passive methods can be realised with relatively simple and inexpensive equipment, but they are not as accurate as active methods, [Raja and Fernandes, 2007]. *Shape from shading*, *photometry*, *shape from focus* and *stereophotogrammetry* are the representatives of this group.

Shape from shading

Within this class of methods, a 3D shape of an object is reconstructed based on a single 2D image. Present shadings and variations in the brightness in the image are analysed and used to determine the depth of single pixels. In order to reconstruct a 3D object, uniform reflectance is assumed. Even though this approach is relatively old [Horn and Brooks, 1989], the improvements in the field of retrieving the 3D geometry out of a single image are ongoing, [Zhang et al., 2015].

Photometry

In this approach a single static camera makes a collection of 2D images, where each of the images is captured under different illumination. Since the amount of reflected light is dependent on the orientation of the surface, surface normals at each point can be recovered.

Shape from focus

Similar as in the photometric approach, methods based on the shape-from-focus rely on processing multiple images under varying conditions. Instead of varying the lighting conditions, shape-from-focus methods vary the distance between the 3D object and a camera. Based on the given distance a focus-based measure is calculated for each pixel of the image. By observing maximal values (according to the defined measure) of each pixel, 3D geometry can be retrieved.

Stereophotogrammetry

In order to reconstruct a 3D object via stereophotogrammetry, multiple images from different directions are captured. On each of the images common points are identified. Knowing the camera position for each image, a line between the camera and a point on the surface of the object can be drawn. The location of the point in a 3D space is determined by the intersection of the corresponding lines. This technique is also known as shape from stereo, stereovision or photometric stereo.

Triangulation

This method relies upon known position of a light source (typically laser) and a camera, as well as the angles between them and a 3D object. Laser light is focused and projected at a previously determined angle on the surface of a 3D object. The reflected light is captured by the camera at a specific angle. Based on the known angles and the distance between the source and the camera, the position of the object surface can be determined.

Time of flight

Time of flight methods measure the elapsed time that light (typically generated by a laser) needs to travel from the light source to the object and back. The distance between the object and the laser corresponds to approximately one half of the distance passed by the laser pulse.

Structured light

In this approach, a pattern of light is projected on the 3D object at a certain, previously determined angle. The surface of the object reflects an image of the pattern. The captured image is in the end analysed and the coordinate points in the 3D space are restored. Geometry of the pattern can be a single point, a line or a more complex 2D grid.

Interferometry

Interferometry approach uses several similar patterns of a structured light. The patterns are principally the same, only slightly displaced or rotated. Once the patterns have been projected on the surface of the 3D object, they result in shadow Moiré effects. The contour lines caused by the Moiré effects are captured by the camera. The height of the object can be restored by calculating the distance between the lines.

Coherent laser radar

This approach combines two signal measurements in order to retrieve the geometry of an object. A focused laser beam is emitted toward a 3D object and the reflected light is coherently processed by a sensor. As the laser light travels between the object and the sensor, it also travels through a reference path of a calibrated optical fibre. The two paths are afterwards combined to determine the absolute distance from the object. With these methods even large scale 3D objects from a far distance (above 10m) can be digitized.

2.2 X-ray based reconstruction

Upon data acquisition, a reconstruction of the three dimensional data can be performed. In this section, an X-ray based reconstruction will be set in focus.

A starting point of an X-ray based reconstruction is a set of projections that result from a partial absorption of X-rays. One projection is obtained for each direction from which the X-rays were emitted. The main reconstruction goal is to retrieve a voxelised matrix (three dimensional array or 3-tensor). Each voxel (three dimensional generalisation of a pixel) contains an attenuation coefficient that characterises the material density of this particular volume portion.

Before describing the details of different reconstruction algorithms, basic terminology will be explained. *Radiography* is a general term for imaging based on X-rays. Even though tomography is often associated with radiography, those two terms are not directly connected. *Tomography* is a general term for imaging in which slices (sections) are created by any kind of transmissive waves. While radiographs (also called projections) represent accumulated absorptions along X-ray lines, tomograms (outcomes from tomographic imaging) represent slices through the original 3D body, see Figure 2.2.

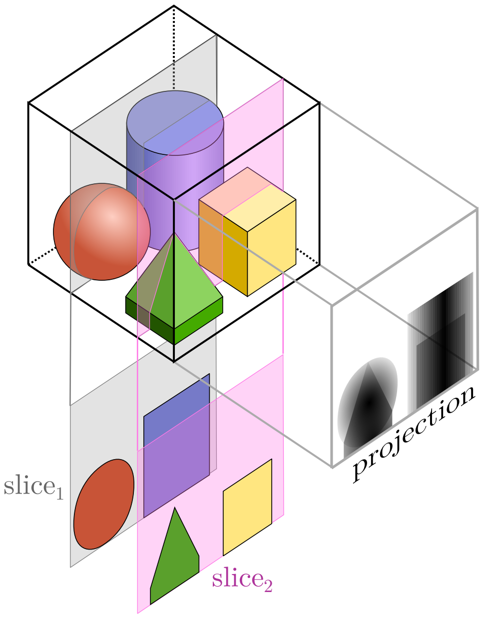


Figure 2.2: Comparison between tomograms (slices) and radiographs (projections).

Regarding the X-ray geometry set-up, there are four different constellations: pencil-beam, parallel-beam, fan-beam and cone-beam (depicted in Figure 2.3). Pencil-beam geometry results from an X-ray source that emits one single ray. Extending this concept with a simultaneous emission of multiple rays along either one or two axes, parallel-beam geometry is obtained. Fan-beam geometry, similar to the pencil-beam, results from an X-ray emission out of a single point. Instead of emitting a single ray, multiple rays within one plane and under certain angle are emitted. In this case, a set of possible rays forms a fan-like shape. Cone-beam geometry is realised by extending this concept in a 3D-space by allowing multiple planes. Here, a set of all possible rays forms a cone.

According to different X-ray acquisition types and geometries, volumetric radiographic imaging can be classified into tomosynthesis, computed tomography (CT) and cone-beam radiography. For each of these subclasses, different reconstruction algorithms come into play. They can be classified into analytical, algebraic, statistical and direct-summation-based.

Tomosynthesis

Tomosynthesis is a tomographic technique in which slice images are obtained after emitting X-rays in very limited rotation angles. Its main application is breast cancer screening (mammography). Modern (digital) tomosynthesis developed from conventional X-ray tomography (also known as blurring-tomography), which has enabled the exposure of only one slice through the body. During the exposure, the

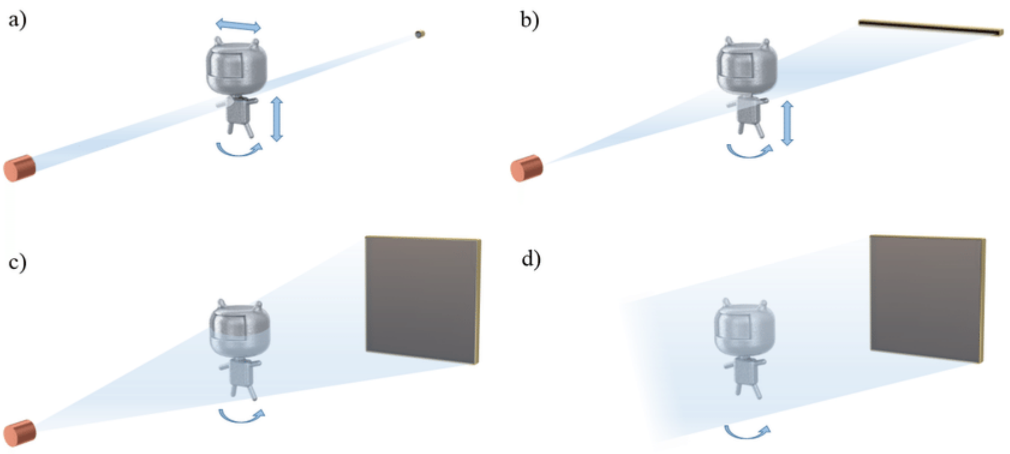


Figure 2.3: Overview of different X-ray geometry set-ups: pencil-beam (a), fan-beam (b), cone-beam (c) and parallel-beam (d), taken from [Jansson, 2017].

X-ray source and the detector are moving on parallel planes in opposite directions and result in multiple projections. Upon obtaining all projections, they are overlain, i.e. the attenuation coefficients are summed up. The corresponding rotational point determines the imaging slice (focal plane). Objects that lie within this slice are always displayed on the same spot of the detector and therefore appear to be sharp, whereas the objects outside the slice appear to be blurred, [Dobbins and Godfrey, 2003].

Tomosynthesis (proposed by [Grant, 1972]) extends this concept by imaging multiple slices. A set of radiographs is made during a single movement of an X-ray source. Different slices can be reconstructed by adding and shifting obtained radiographs. In addition to this summation reconstruction approach, algorithms for reconstruction similar to CT reconstruction techniques (in detail presented in the sections below) can be applied as well. However, they are enhanced with deblurring algorithms, which are necessary due to the limited scanning angle (up to 60°), [Gomi, 2016], [Dobbins and Godfrey, 2003].

Computed Tomography

Computed tomography (CT) is another tomographic technique for obtaining slice images of a 3D body via X-rays. In order to reconstruct a single slice, measurements from various directions of at least 180° rotational range are necessary. Based on Radon's and Cormack's theoretical work, Hounsfield conducted the first CT apparatus in 1972, [Kalender, 2011]. Since then, the CT has been rapidly developed and nowadays represents a standard tool in various medical disciplines.

In CT, an X-ray source and a detector are positioned at opposite sides of the body and translate and/or rotate synchronously around it. The development of CT can be historically classified in four generations. The first generation of CT was characterised by multiple translation, followed by rotation of a pencil-beam. The second generation CT apparatus used the same approach for small fan-beams, where a fan beam is positioned orthogonally to the rotational axis. A full 360° continuous rotation on a circular path of a big fan-beam is representative of the third (featuring rotating detector) and the fourth (featuring stationary detector) generation, [Dössel and Buzug, 2014].

Most of CT apparatuses nowadays are representatives of the third generation, meaning that the slice-wise reconstruction is obtained via fan-beam with a rotating detector. Many further enhancements have been developed, such as: dual-energy CT (material-selective imaging), dynamic CT (imaging of temporal changes) and dual-source CT (increasing power), [Kalender, 2011].

Usually utilised reconstruction methods in modern CT are analytical methods, mainly filtered back-projection. Iterative (statistical and algebraic) methods lost their popularity in the last decades due to the high computational costs, which has resulted in time-consuming reconstruction, [Kalender, 2011]. However, constant rapid development of computer components reopens the door for their further development. Recently, dose-reduction orientated iterative methods (such as ASIR, IRIS, MBIR, AIDR and SAFIRE) have been discussed, [Qiu and Seeram, 2016].

Single-slice (2D) scanning can be extended on volume (3D) scanning by using spiral (helical) CT. In this approach, a 3D body is moved along one axis, while the X-ray source and the detector are (the same as before) continuously rotating on a circular path. Another way how to reconstruct a 3D volume is by using cone-beam radiography.

Cone-beam radiography

Cone-beam radiography is also known as cone-beam CT (CBCT), cone-beam volume CT, multiple-slice CT and flat-panel-detector CT. The underlying procedure is similar to the conventional CT, but with cone-beam replacing the fan-beam geometry. Even though a 3D reconstruction can be obtained by a subsequent 2D reconstruction of slices and stacking them together, a direct 3D reconstruction via cone-beams reduces the data collection time, [Kak and Slaney, 2001]. The application of cone-beam radiography is versatile: medicine (dentistry, orthopaedics and interventional radiology), industrial material inspection and general digitization of 3D objects. For the majority of medical applications the X-ray source and the detector rotate around the body. In the industrial applications however, a 3D object is placed on a rotating table, while the X-ray source and the detector remain steady, [Dössel and Buzug, 2014], [Louis, 2016].

Reconstruction algorithms for cone-beam radiography are essentially 3D generalisations of 2D single-slice CT algebraic and analytical algorithms. While the 3D extension for the algebraic algorithms is straight forward, for the analytical filtered back-projection this is more complicated. The Feldkamp algorithm, a 3D extension of filtered back-projection, is the most widely used cone-beam reconstruction algorithm, [Sun et al., 2012], [Kalender, 2011].

> Cone-beam radiography has a smaller dynamic imaging range than the conventional CT. In general, cone-beam reconstructions suffer from more shading cupping artefacts, reduced accuracy and deficient low-contrast sensitivity, [Shaw, 2014]. This disables capturing small density differences and results in a coarse material differentiation, [Dössel and Buzug, 2014].

2.2.1 Forward-projection: Radon transform

As an X-ray traverses through an object under examination, it attenuates due to the absorption of the objects material. For an X-ray with initial intensity I_0 that passes through an object out of material with attenuation coefficient μ and thickness s, intensity I after the attenuation is characterised by the Beer-Lambert law:

$$I = I_0 e^{-\mu s} \tag{2.1}$$

If an X-ray is traversing through multiple volume elements with different attenuations μ_i with corresponding thicknesses s_i; intensity I is given by:

$$I = I_0 e^{\sum -\mu_i s_i} \approx I_0 e^{-\int \mu(s) ds} \tag{2.2}$$

Since the modelled solution data is in a voxelised form of unknown density (given through the *attenuation coefficients*), we can assign the value μ_i to each voxel. The intensity values after the attenuation are given by the projection data, i.e. by the pixel values on the radiograph, which corresponding X-ray intersects. Initial intensity values are typically measured by performing an "empty scan", i.e. measuring pixel values on a radiograph, which was created without the object. With known I, I_0 and voxel sizes s_i (which approximate the ray length within the voxel), a relation between voxel densities μ_i can be obtained as in Equation (2.3). Deriving the voxel densities based on the measured intensities is the main task of the reconstruction.

$$\int \mu(s) ds \approx \sum \mu_i s_i = \ln \frac{I_0}{I} \tag{2.3}$$

In order to obtain a simple model of the projection data (forward projection), a 2D case (parallel-beam or fan-beam geometry) will be considered. A 3D case on cone-beam geometry will be covered in Section 2.2.4.

Each X-ray line can be represented by two parameters r and ϕ. While ϕ is the angle between the x-axis and the normal vector of the line, r is the orthogonal distance from the line to the origin. With that, each of the X-ray lines can be described as a set of points $\left(\begin{smallmatrix} x \\ y \end{smallmatrix}\right)$ with:

$$x \cos(\phi) + y \sin(\phi) = r \tag{2.4}$$

For a fixed $r \in \mathbb{R}$, consider the line $\{\binom{r}{t} \mid t \in \mathbb{R}\}$. It is a line which is parallel to the y-axis and has the signed distance r to it. If this vertical line is rotated around the origin for the angle ϕ, the line $\{\binom{x}{y} \mid x\cos(\phi) + y\sin(\phi) = r\}$ is obtained and hence it can be parametrised as:

$$\begin{pmatrix} x(t) \\ y(t) \end{pmatrix} = \begin{pmatrix} \cos(\phi) & -\sin(\phi) \\ \sin(\phi) & \cos(\phi) \end{pmatrix} \begin{pmatrix} r \\ t \end{pmatrix} = \begin{pmatrix} r\cos(\phi) - t\sin(\phi) \\ r\sin(\phi) + t\cos(\phi) \end{pmatrix} \tag{2.5}$$

With this representation, attenuation along an X-ray line leads to:

$$b_\mu(\phi, r) := \int\limits_{x\cos(\phi)+y\sin(\phi)=r} \mu(x, y)ds \approx \int\limits_{-\infty}^{\infty} \mu(r\cos(\phi) - t\sin(\phi), r\sin(\phi) + t\cos(\phi))dt \tag{2.6}$$

The transformation from the function $\mu(x, y)$, defined on $\mathbb{R}^2$, to function $b_\mu(\phi, r)$, defined on $[0, 2\pi) \times \mathbb{R}$, is known as Radon transform. Radon was the first one to solve the reconstruction problem out of the line integrals defined in Equation (2.6) in 1917, [Birkfellner, 2015]. The Radon transform is in general used to describe the attenuation of X-rays along the ray path, resulting in a forward-projection. Calculating the inverse Radon transform, i.e. the back-projection is a problem tackled by different reconstruction algorithms.

Restricted by the scope of application in this thesis, reconstruction algorithms can be divided into analytical and algebraic algorithms, depending on how the density function is being determined. While algebraic algorithms consider unknown density to be a discrete point function, analytical algorithms consider it to be a continuous function and discretize the solution at the very end. The analytical determination of the density function is done by solving continuous integral equations. The algebraic approach lies, however, in iterative solving of a system of equations, [Kalender, 2011].

In the following Sections 2.2.2 and 2.2.3 a 2D reconstruction of one slice, as it is applied in the conventional CT, via algebraic and analytic methods will be presented. A generalization of these algorithms in a 3D space on a cone-beam geometry follows in the Section 2.2.4.

2.2.2 Algebraic 2D reconstruction

The starting point in the algebraic reconstruction approach is a discretisation of the unknown densities. In a 2D space this results in a square grid, where each of the unknown densities $\mu(x, y)$ represents a constant value assigned to one cell. For the sake of simplicity, consider a predefined order of pixels and let μ_i denote the unknown density of the i-th cell, where $i = 1, 2, ..., n$. In this algebraic approach an X-ray is modelled as a line with a certain thickness. Within a parallel-beam geometry X-rays are modelled as stripes with constant width, whereas within a fan-beam geometry they are modelled as elongated triangles. With altogether m rays, one can define (weighting) factors A_{ji}, which describe the surface ratio between the area covered by the j-th ray within i-th cell and the total area of the i-th cell. If b_j denotes the discretized Radon transform along the j-th ray, the relationship between unknown densities and known Radon transforms can be expressed as in Equation (2.7). An illustration of this concept for a fan-beam geometry is depicted in Figure 2.4.

$$\sum_{i=1}^{n} A_{ji}\mu_i = b_j, \qquad j = 1, \dots, m \tag{2.7}$$

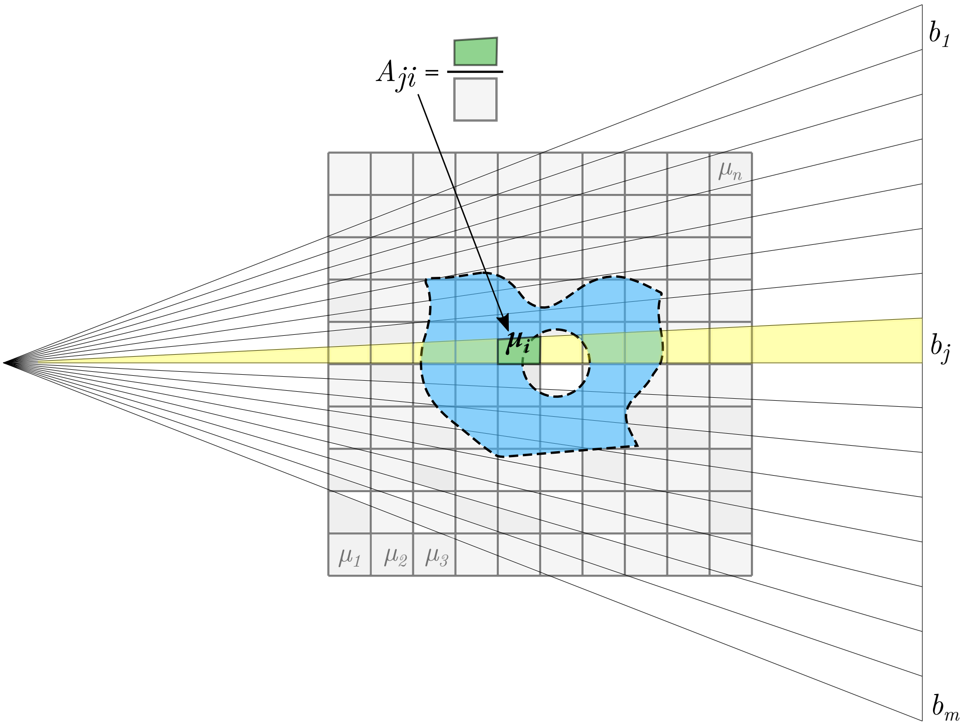


Figure 2.4: Discretized algebraic model for a fan-beam geometry.

Equation (2.7) can be expanded in a set of m equations, i.e. one equation for each X-ray:

$$\begin{aligned} A_{11}\mu_1 + A_{12}\mu_2 + \cdots + A_{1n}\mu_n &= b_1 \\ A_{21}\mu_1 + A_{22}\mu_2 + \cdots + A_{2n}\mu_n &= b_2 \\ &\vdots \\ A_{m1}\mu_1 + A_{m2}\mu_2 + \cdots + A_{mn}\mu_n &= b_m \end{aligned} \tag{2.8}$$

The set of Equations (2.8) can be simply rewritten as a matrix-vector multiplication. Since the geometry set-up ($m \times n$ matrix $\mathbf{A}$) and Radon transforms (vector $\mathbf{b}$ of length

m) are known prior to reconstruction, unknown densities (vector $\boldsymbol{\mu}$ of length n) can be obtained by solving:

$$\mathbf{A}\boldsymbol{\mu} = \mathbf{b} \tag{2.9}$$

However, this approach (explicit calculation of the matrix) has several drawbacks and one of them is high memory demand concerning the storage of the matrix. For reconstructing moderately sized grid with 180 × 180 unknowns and only 180 rotational steps, a matrix would have the following size:

$$(180 \cdot 180) \times (180 \cdot 180) = 32400 \times 32400.$$

The storage consumption of such a matrix filled with doubles (1 double = 8 bytes) would correspond to 7.8 gigabytes. Handling these large systems by conventional methods for matrix inversion is time consuming even for modern computers, [Birkfellner, 2015]. Moreover, this big matrix is sparse, since each of the X-rays passes through only a small number of cells. This motivates a new, iterative approach without explicit matrix formation for solving the system of Equations (2.8).

An iterative approach for solving this problem was first proposed by [Kaczmarz, 1937]. In this approach, vector $\boldsymbol{\mu}$ is considered to be a point in an n-dimensional space. Each of the equations in (2.8) represents a (hyper)plane. If there is a unique solution of the system, all planes are intersecting in a single point. Starting from initial value $\boldsymbol{\mu}^0$, each of the following intermediate solutions $\boldsymbol{\mu}^j$ is given by projecting $\boldsymbol{\mu}^{j-1}$ on the j-th plane (equation). Expressed with a formula with $\mathbf{A}_j$ representing the j-th row of a matrix, this procedure leads to:

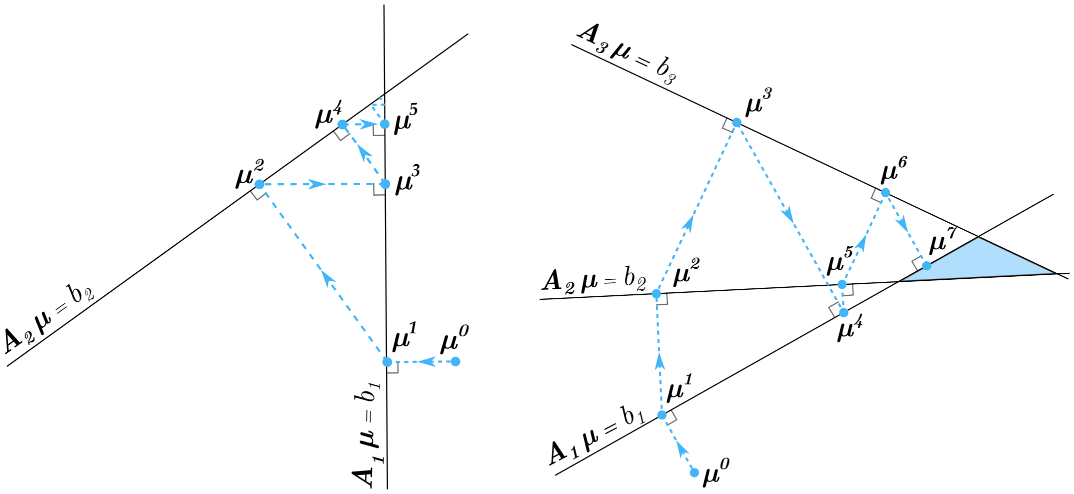
$$\boldsymbol{\mu}^j = \boldsymbol{\mu}^{j-1} - \frac{\boldsymbol{\mu}^{j-1} \cdot \mathbf{A}_j - b_j}{\mathbf{A}_j^T \cdot \mathbf{A}_j} \mathbf{A}_j \tag{2.10}$$

If there exists a unique solution for the system of Equations (2.8), i.e. $n = m$ and $\mathbf{A}$ has full rank, [Tanabe, 1971] proved that the intermediate solutions $\boldsymbol{\mu}^j$ converge to it (see Figure 2.5a). However, faster convergence is achieved if the (hyper)planes are orthogonal to each other, [Kak and Slaney, 2001].

In a common situation in X-ray based reconstruction there are much more X-rays (equations or (hyper)planes) than the unknowns, i.e. an overdetermined system ($m > n$). In this case, there exists no unique solution, but the intermediate solutions will approach to the intersection area of the planes, as depicted in the Figure 2.5b.

In an undetermined system ($m < n$) there is no unique solution, but an infinitive number of possible solutions. [Tanabe, 1971] proved that the proposed iterative approach converges to the most optimal achievable solution concerning the chosen initial guess.

Based on this algorithm, three different algebraic reconstruction techniques will be further discussed: Algebraic Reconstruction Technique (ART), Simultaneous Iterative Reconstruction Technique (SIRT) and Simultaneous Algebraic Reconstruction Technique (SART).



(a) In a system of equations (planes) with a unique solution the projected intermediate solutions converge to the plane intersections.

(b) In an overdetermined system of equations the projected intermediate solutions converge to the intersection area.

Figure 2.5: Karczmarz algorithm.

ART

Algebraic Reconstruction Technique simplifies the Karczmarz approach by omitting the calculation of weighting factors A_{ji}. Instead, the total number of the intersected cells for each ray (N_j) is obtained. If the centre of the i-th cell lies within the j-th ray, corresponding A_{ji} is set to 1, otherwise on 0. With that, the denominator from the Equation (2.10) equals to:

$$\mathbf{A}_j^T \cdot \mathbf{A}_j = \sum_{k=1}^{n} A_{jk}^2 = N_j \tag{2.11}$$

The minuend of the numerator from Equation (2.10) can be perceived as a virtual projection (v_j) of the j-th ray, obtained by transmissing through the current intermediate solution:

$$v_j = \boldsymbol{\mu}^{j-1} \cdot \mathbf{A}_j \tag{2.12}$$

With the Equations (2.10), (2.11) and (2.12), correction of the single i-th cell by the j-th ray in ART is given by:

$$\mu_i^j = \mu_i^{j-1} - \frac{v_j - b_j}{N_j} A_{ji} \tag{2.13}$$

With this, each of the iterations of an ART ray-based approach can be summarised as follows:

1. For each ray, find the corresponding intersected cells.
2. For each ray, calculate the virtual projection.
3. For each ray, calculate the cell-correction (normalised difference between the virtual and the actual projection).
4. For each intersected cell, apply the correction.

SIRT

The SIRT approach is very similar to ART, but with one difference: calculated corrections are not immediately applied for the update of the corresponding intersected cells. Instead, those intermediate-corrections are accumulated in an additional (correction) matrix. Furthermore, a total number of the rays passing through the i-th cell is precomputed in the variable M_i. With that, an average cell correction is calculated for all rays. The cell update is therefore applied only after processing all the rays. With that, one SIRT-iteration has been accomplished and can be summarised as follows:

1. For each ray, find the corresponding intersected cells.
2. For each ray, calculate the virtual projection.
3. For each ray, calculate the cell-correction (normalised difference between the virtual and the actual projection) and store it cumulatively.
4. For each cell, calculate the total number of rays passing through it and store it.
5. After processing all the rays, update each cell with the average cell-correction.

An update of a single i-th cell in the k-th iteration is given by the following equation:

$$\mu_i^k = \mu_i^{k-1} - \frac{\sum_{j \in I_i} \left(\frac{v_j - b_j}{N_j} A_{ji} \right)}{M_i} A_{ji}, \tag{2.14}$$

with I_i being the set of all the rays that intersect the i-th cell.

SART

Unsmooth reconstructions resulting from ART can be enhanced by SIRT approach. However, both ART and SIRT are prone to salt and pepper noise due to the imprecise modelling of the weighting factors, [Kak and Slaney, 2001]. This leads to SART - an enhanced algebraic reconstruction approach, which was proposed by [Andersen and Kak, 1984]. Nowadays it is one of the standard X-ray based reconstruction techniques.

The same like in SIRT, a simultaneous correction of all cells is carried out in the end of one iteration. However, modelling of the forward projection in SART differs immensely. Unknown density is not assigned to each cell, but to the cell centre points. Each X-ray is modelled as a line, and not as an elongated triangle. The X-ray integral (virtual forward projection v_j) is then calculated as a sum of interpolated weighted densities. X-ray paths are equidistantly sampled with K_j points, with Δp being the distance between each two points (Δp is typically set to the half of the cell size):

$$v_j = \sum_{k=1}^{K_j} \mu(x_{jk}, y_{jk}) \Delta p \tag{2.15}$$

Let the weighting factors d_{ijk} stand for the bilinear contribution of the i-th cell to the k-th sample point on the j-th ray:

$$d_{ijk} = \max\{0, 1 - \tfrac{|x_{jk}-x_i|}{\Delta s}\} \cdot \max\{0, 1 - \tfrac{|y_{jk}-y_i|}{\Delta s}\}, \tag{2.16}$$

where (x_i, y_i) is the center of the i-th cell and Δs the width/height of a single pixel. In particular, $\sum_{i=1}^{n} d_{ijk} = 1$. The density value in a single point is then given by bilinear interpolation of the four neighbouring points μ_i:

$$\mu(x_{jk}, y_{jk}) = \sum_{i=1}^{n} d_{ijk} \mu_i, \qquad k = 1, ..., K_j \tag{2.17}$$

Combining Equations (2.15) and (2.17), forward projection v_j can be approximated as:

$$v_j = \sum_{k=1}^{K_j} \sum_{i=1}^{n} d_{ijk} \mu_i \Delta p = \sum_{i=1}^{n} \left(\sum_{k=1}^{K_j} d_{ijk} \Delta p \right) \mu_i =: \sum_{i=1}^{n} A_{ji} \mu_i \tag{2.18}$$

The sampling coefficients A_{ji} are therefore proportional to the sum of the weighting factors made by the i-th cell to the all sampling points along the j-th ray. Even though the sampling coefficients have the same notation as the matrix coefficients in ART and SIRT, they represent different things. With this, each of the SART iterations can be summarised as follows:

1. For each ray, calculate the virtual projection by bilinear sampling,
2. For each ray, calculate the sum of the weighting factors made by all cells.

3. For each ray, calculate the cell-correction (normalised difference between the virtual and the actual projection) and store it cumulatively.

4. For each cell, calculate the sum of sampling coefficients over the rays that influences it.

5. After processing all of the rays, update each cell with the weighted cell-correction.

Update of a single i-th cell in k-th iteration can be expressed by the following equation:

$$\mu_i^k = \mu_i^{k-1} - \frac{\sum\limits_{j \in I_i} \left(\frac{v_j - b_j}{\sum\limits_{i=1}^{n} A_{ji}} A_{ji} \right)}{\sum\limits_{j \in I_i} A_{ji}} A_{ji}, \tag{2.19}$$

with I_i being the set of all the rays that influence the i-th cell.

In practice, significant improvements in the reconstructed data are achieved by using up to five iterations, [Tita, 2007]. A further SART improvement of the noise reduction is the introduction of a relaxation factor and Hamming window. A relaxation factor ($\in [0, 1]$) is set to one for the first iteration and declines with each iteration. With that, corrections of the increasing iterations (possible noise) are weakened. With a (longitudinal) Hamming window, weighted corrections are emphasised on the central portion of the rays, since the object to reconstruct is typically positioned around the centre of a square grid, [Kak and Slaney, 2001].

2.2.3 Analytical 2D reconstruction

Analytical reconstruction methods assume an object density function $\mu(x, y)$, as well as the measured projection data (Radon transform) b_μ to be continuous. Solving the reconstruction problem in that case means solving a continuous integral equation. This chapter represents the 2D analytical reconstruction for a parallel-beam geometry. Fan-beam geometry can be easily transformed into parallel-beam geometry via rebinning (resorting of the rays), [Kalender, 2011].

Direct Fourier method

In order to obtain the solution of an analytical reconstruction problem, a connection between the Fourier transform and Radon transform is necessary. Fourier transform and its inverse for functions in one variable is given in the Equation (2.20) and in two variables in the Equation (2.21). Within this thesis, the Fourier transform of a general function f will be denoted by its corresponding capital letter F.

$$\begin{aligned} F(k) &= \int_{-\infty}^{\infty} f(x) e^{-2\pi i k x} dx \\ f(x) &= \int_{-\infty}^{\infty} F(k) e^{2\pi i k x} dk \end{aligned} \tag{2.20}$$

$$\begin{aligned} F(k,s) &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y) e^{-2\pi i (kx+sy)} dxdy \\ f(x,y) &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F(k,s) e^{2\pi i (kx+sy)} dkds \end{aligned} \tag{2.21}$$

A connection between the Radon (b_μ) and Fourier transforms for the density function μ is given by the projection slice theorem. For a simpler understanding, the projection slice theorem will be first carried out for lines without rotation ($s = 0$). Starting from the 2D Fourier transform in Equation (2.21), the Fourier transform of the density function $\mu(x, y)$ can be expressed as:

$$\begin{aligned} M(k,0) &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \mu(x,y) e^{-2\pi i k x} dxdy = \int_{-\infty}^{\infty} \left(\int_{-\infty}^{\infty} \mu(x,y) dy \right) e^{-2\pi i k x} dx \\ &= \int_{-\infty}^{\infty} b_\mu(0,x) e^{-2\pi i k x} dx = B_\mu(0,k), \end{aligned} \tag{2.22}$$

where M is the 2D Fourier transform of μ and B_μ of b_μ from Equation (2.6).

A geometrical interpretation of the obtained equality in the Equation (2.22) is the following: taking a single slice at $s = 0$ of a 2D Fourier transform is the same as computing the 1D Fourier transform of the Radon transform for $\phi = 0$. Generalisation of this concept to all the angles leads to the projection slice theorem (also known as Fourier slice theorem), [Cramér and Wold, 1936]:

$$M(r\cos(\phi), r\sin(\phi)) = B_\mu(\phi, r) \tag{2.23}$$

With this theorem, direct Fourier method for reconstructing the density function out of the known Radon transforms can be provided in only two steps:

1. For different angles, calculate 1D Fourier transforms of the known Radon transforms.

2. Calculate the 2D inverse Fourier transform of the obtained result.

However, realisations of this approach lead to quite noisy reconstructions, [Birkfellner, 2015]. From Equation (2.23) it is clear that the reconstruction is possible only along the radial lines. This implies high inaccuracies far away from the origin. A different reconstruction algorithm, based on the projection slice theorem as well, is called the filtered back-projection.

Filtered back-projection

Filtered back-projection algorithm, also known as convolution-back-projection, was proposed by [Bracewell and Riddle, 1967]. It is still very popular (in fact the most frequently used algorithm) for the X-ray based reconstruction in (single-sliced) CT due to its speed and robustness [Kalender, 2011].

Starting from the 2D inverse Fourier transform, it can be shown that the following holds, [Birkfellner, 2015]:

$$\begin{aligned}\mu(x,y) &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} M(k,s)e^{2\pi i(kx+sy)}dkds \\ &= \int_{0}^{\pi}\int_{-\infty}^{\infty} M(r\cos(\phi), r\sin(\phi))e^{2\pi ir(x\cos(\phi)+y\sin(\phi))}|r|drd\phi\end{aligned} \tag{2.24}$$

Applying the projection slice theorem on Equation (2.24) and inserting $t = x\cos(\phi) + y\sin(\phi)$, the filtered back-projection can be expressed as follows:

$$\mu(x,y) = \int_{0}^{\pi}\left(\int_{-\infty}^{\infty} B_\mu(\phi,r)|r|e^{2\pi irt}dr\right)d\phi \tag{2.25}$$

The expression in the brackets in Equation (2.25) is in fact the inverse Fourier transform of the function $B_\mu(\phi,r)|r|$ evaluated at t. It can be interpreted as a high-pass filtering operation ($filt$) of the Radon transform b_μ, achieved as a multiplication in the Fourier space (also known as convolution, denoted by $*$).

With that, the value $\mu(x,y)$ can be obtained by deriving $b_\mu * filt$ evaluated at t for all the angles and finally integrating (adding) them:

$$\mu(x,y) \approx \sum_{k} \left(b_\mu(\phi_k) * filt\right)(t_k) \tag{2.26}$$

One (angular) step of the filtered back-projections for the reconstruction of a 2D slice via parallel rays can be summarised as follows:

1. Create a filter and apply Fourier transform to it.
2. For all the parallel rays with ϕ_k, apply Fourier transform to the Radon transform of the density function.
3. Multiply the filter with the Radon transform.
4. Apply the inverse Fourier transform to the result.
5. Smear the obtained values along the corresponding ray lines.

2.2.4 Cone-beam 3D reconstruction

Cone-beam 3D reconstruction can be in general classified into theoretically exact and approximate reconstruction. In order to perform exact reconstruction, the Tuy condition (also known as Tuy-Kirillov condition) has to be fulfilled. The Tuy condition requires that every plane intersecting the object must intersect the trajectory of movement of the X-ray source and the detector at least once, [Tuy, 1983].

In most industrial applications the X-ray source and the detector are moving on a circular trajectory around the object, [Louis, 2016]. In that case, only central plane (mid-plane which is parallel to the plane where the trajectory lies) of the object intersects the trajectory of movement and the Tuy condition remains unfulfilled. High reconstruction quality can be ensured only for the central plane and the structures that are more far away from the central plane will be in general of poorer quality than the closer ones. Resulting artefacts are known as cone-beam artefacts, [Kalender and Kyriakou, 2007]. Low-contrast resolution and general reconstruction precision is reduced due to these artefacts, [Shaw, 2014]. Small density differences are therefore hard to distinguish and cone-beam 3D reconstruction based on the circular trajectories results in a coarse material differentiation.

Exact cone-beam reconstruction based on other approaches (such like non-circular trajectories) is an ongoing research, [Katsevich et al., 2017], [Louis, 2016], [Varslot et al., 2011].

In this section, two commonly used approximation algorithms based on circular trajectories of an X-ray source and a flat detector will be presented: SART and Feldkamp algorithm.

SART

A 3D (cone-beam) generalisation of the 2D-SART, which was presented in the Section 2.2.2, is straight-forward. The same holds for other reconstruction techniques that are members of the algebraic reconstruction family. For the parallel-beam, the coordinates of the sample points are directly given. For the cone-beam however, sample points along a ray are obtained by scaling the end-points of a corresponding X-ray (i.e. detector points) according to the beam geometry. Bilinear interpolation in the sample points can than be replaced by a trilinear interpolation. Different distances covered by the X-rays within a cone-beam setting with flat detectors have to be taken into account as well.

With that, each of the 3D-SART iterations can be summarised as:

1. For each ray, calculate the virtual projection according to cone-beam geometry by trilinear sampling.
2. For each ray, calculate the sum of the weighting factors made by all cells.
3. For each ray, calculate the cell-correction (normalised difference between the virtual and the actual projection) and store it cumulatively.
4. For each cell, calculate the sum of sampling coefficients over the rays that intersect it.
5. After processing all of the rays, update each cell with the weighted cell-correction according to cone-beam geometry.

Feldkamp algorithm

Due to the circular (planar) trajectory of the X-ray source and detector around a 3D object, the complete set of projection data is not available. In fact, this is only the case for the central plane. Line projections for all the angles around the central plane normal are given. A 3D straight-forward generalisation of the filtered back-projection with (exact) inverse Radon approach is therefore not applicable.

An approximation algorithm for the 3D analytical reconstruction is the Feldkamp algorithm, also known as Feldkamp-Davis-Kress (FDK) algorithm. It was proposed by [Feldkamp et al., 1984] and it is still widely applied, [Louis, 2016]. It assumes that the X-ray source is sufficiently far away from the object, so that the cone-opening angle is relatively small. This leads to an approximation of the original geometry by neglecting the cone-opening angle. With that, slightly tilted planes become planes parallel to the central plane and for each of them inverse Radon approach can be applied. The 3D reconstruction is then achieved by applying weighting on the slice-wise obtained results from the 2D filtered back-projection algorithm, as previously presented.

One (angular) step of the 3D reconstruction with the Feldkamp algorithm can be summarised as follows:

1. Collect filtered projection data for all detector lines:
    a) Create a filter and apply Fourier transform to it.
    b) For all rays, apply Fourier transform to the Radon transform of the density function.
    c) Multiply the filter and the Radon transforms.
    d) Apply inverse Fourier transform to the result.
2. Apply weighted back-projection of the obtained values for all slices (i.e. almost parallel planes).

2.3 Cone-beam X-ray apparatus

2.3.1 Introduction

Even though the first clinical CTs appeared on the market in the 1970s, it lasted further 20 years for the first industrial X-ray apparatus for examining technical objects to appear. With this breakthrough, it was possible for the first time to inspect the internal structures of objects, such as cavities, inclusions and material formations.

The main components of the most (industrial) X-ray apparatuses are an X-ray source, a flat detector and a rotary axis on which objects are fixed. A linear axis is optionally used for the free positioning of the object between the source and the detector and hence provides magnification of the object. In the majority of medical applications, the X-ray source and detector are moving around the stationary patient. The industrial X-ray apparatuses differ in that the radiation source and the detector remain stationary while the object rotates. Although there is in principle no difference whether the source and detector, or the measured object rotate, the latter concept is considered to be more advantageous. Rotary axes can be produced with high accuracy and cost-efficiently. However, if the X-ray source and the detector have to be mounted and accelerated, the technical requirements for accuracy and stability, as well as production costs, are increased, [Kalender, 2011].

Regarding the beam geometry and the detector shape, a cone-beam geometry with the corresponding flat detectors are typically used. The main advantage, when compared to the fan-beam geometry with 1D detector lines, is the reduction of the measuring time since multiple slices can be scanned at once, [Christoph and Neumann, 2012].

While in medical applications X-ray dose reduction is a big concern, in industrial application it does not play a significant role (as long as the materials are X-ray invariant). For this reason, higher radiation intensities for inspecting technical objects are applicable. While the acceleration voltage in medical diagnostics ranges between 25 and 150 kV, and in radiation therapy between 10 and 300 kV, in industrial applications, it goes up to 500 kV, [Christoph and Neumann, 2012]. To protect the users from radiation, X-ray apparatuses are typically built as enclosed machines with shields made of several lead layers. With that, the users sustain the negligible amount of radiation.

Various physical effects can influence the reconstructed results. Regarding the measurement errors, among the most influential ones are beam hardening, scattered radiation and cone-beam artefacts, [Christoph and Neumann, 2012].

2.3.2 Scanning procedure

The first step of the scanning procedure via a cone-beam X-ray apparatus is the object fixation. If the objects are not fixed, significant measurement deviations can occur. Each object can be fixed either directly on the rotatory axis or on the rotatory table. The fixation material should be as transparent as possible with respect to the X-ray radiation in order to minimise the influences on the measurement result. While plastic fixation material is typically used for the objects made of metal, a thin foam is commonly used for the objects made of plastic.

After completing the object fixation, X-ray relevant parameters have to be set. These are the acceleration *voltage* of the X-ray tube, the *current* of the X-ray tube and the (optional) exposure (integration) time. With a higher voltage, electrons have higher velocity when they hit the metal target, and the energy of the produced X-ray photons is higher. Therefore, the voltage level determines the hardness of the X-ray radiation. The current, on the other hand, determines the radiation intensity, i.e. the number of accelerated electrons. Those two parameters are relevant for setting the right contrast of the projection image (radiograph). A suitable voltage is set first for each object individually, depending on its thickness and material. The fitting current is set directly afterwards. An approximate guideline for the voltage setting for plastic, aluminium and steel/ceramic regarding different thicknesses is given in the Table 2.1.

In general, projection images should be as bright as possible for obtaining low noise, as long as they are not over-exposed. In order to obtain optimal projection images, additional filters (typically made of aluminium, tin or copper) can be positioned at the output of the X-ray tube. They block low-energy X-rays, which in general do not significantly contribute to the final result.

Table 2.1: An approximate guideline for the voltage setting for objects made of plastic, aluminium and steel/ceramic regarding different thicknesses, taken from [Christoph and Neumann, 2012].

Voltage	130 kV	150 kV	190 kV	225 kV	450 kV
Plastic	< 90 mm	< 130 mm	< 200 mm	< 250 mm	< 450 mm
Aluminium	< 30 mm	< 50 mm	< 90 mm	< 150 mm	< 250 mm
Steel/ceramic	< 5 mm	< 8 mm	< 25 mm	< 40 mm	< 70 mm

Since the object can be of all possible sizes and made of a wide spectrum of single or multiple materials (possibly even unknown to the user), manual setting of the relevant X-ray parameters are necessary. This is typically done in an iterative process, where a test radiograph is made for each set of varying parameters. Optimal parameters are chosen based on the user's personal judgement of the image contrast and optional evaluation of the grey-values distribution in a corresponding histogram. Since good contrast should be ensured for projections from all directions, users typically make several test-radiographs, [Reidl et al., 2016].

After setting the relevant X-ray parameters, reference projections (empty projections without the object) are made. Upon obtaining the reference images, the object is put back on the rotatory axis. Further parameters relevant for the reconstruction have to be chosen, such as the size of the reconstructing space, the voxel resolution and the number of projections. With that, the preparation step is finished and the data acquisition process, followed by the reconstruction, can be performed. This is typically realised as a non-transparent process, which informs users about the reconstruction result only at the very end. Depending on the chosen reconstruction parameters, the data acquisition with reconstruction lasts between 10 and 60 minutes, [Reidl et al., 2016]. Upon successful reconstruction, voxel data is typically transformed into another model representation, such as meshes or point clouds, which serve as a starting point for further post-processing, [Christoph and Neumann, 2012].

2.4 Processing of 3D models

2.4.1 Model representation

3D models can be specified by either *point clouds* (PC) - a set of boundary points, *wire models* - a set of boundary edges, *boundary representations* (B-rep) - a set of boundary faces, *constructive solid geometries* (CSG) - a set of mathematical primitives combined by boolean operators, or *voxel models* - a 3D array of elements (voxels) that contain additional information, such as density or colour.

B-rep models consist of a set of organised faces that define the models' interior and exterior. T-spline surfaces, NURBS surfaces and (polygon) meshes are different types of B-reps. A typical choice for the polygons in mesh representations are triangles and convex quadrilaterals. Triangular meshes have been established as the standardised 3D model representation in the RP environment, [Gebhardt, 2013, Müller et al., 2013], even though the voxel representation generally yields in better quality [Reinhart et al., 2004]. Recently, different authors report using the voxel representation due to the numerous benefits of their deployment in AM, [Tedia and Williams, 2016]. Other model representations are because of that translated into voxel-based representations. In spite of voxelisation, an additional translation into standardised B-rep is conducted prior to fabrication, [Lin et al., 2007].

3D digitizing methods typically result in either voxel models (transmissive methods) or point clouds (remaining methods), [Raja and Fernandes, 2007]. Converting point clouds to meshes can be done by Delaunay triangulation (for convex domains), alpha shapes, ball pivoting (BPA), moving least squares and Poisson surface reconstruction. Voxel-based models that result from the transmissive methods are often stored in a DICOM format - a standard for storing the information in medical imaging. Converting voxel models to corresponding meshes can be realised via the marching cubes algorithm, [Lorensen and Cline, 1987].

Point clouds, meshes, and voxel models only contain local, or *low level* information about the model, such as coordinates, normals, and curvature. However, reverse engineering often seeks to deduce *middle level* information (basic primitive shape types and parameters) and *high level* (constraints such as parallelism, concentricity, symmetry, etc.), [Yi et al., 2014]. This information relays function and design intent and makes the 3D models applicable for design changes or simulations.

2.4.2 Recognition of simple 3D primitives

After successful data acquisition and 3D reconstruction, a low-level model is obtained. To establish middle- and high-level models, further processing is necessary. Many real world objects are modelled with simple 3D primitives, such as planes, spheres and cylinders, [Camurri et al., 2014]. By detecting those simple parametric surfaces, imperfect portions of the reconstructed object can be improved. The surface fitting of general shapes is in general difficult, and users' interaction is often necessary, [Gelfand and Guibas, 2004].

The vast majority of authors report point clouds being the corresponding model representation for the recognition of simple primitive surfaces, as well as general surfaces. A detailed overview of state of the art in surface reconstruction from point clouds that result from 3D scanners is presented in [Berger et al., 2014]. Most approaches for the recognition of simple primitives are based either on RANSAC, Hough Transform or region growing, [Drost and Ilic, 2015].

Region growing methods start from the small portion of the initial points (seeds) and grow them by connecting adjacent segments. With that, the regions that have similar characteristics, such as belonging to the same surface parametrisation, are segmented. The seeds are often set manually, and only one connected region can be detected for each seed.

Hough transform (HT) was introduced by [Hough, 1962] to detect curves in bubble chamber photographs. HT is nowadays commonly used to detect straight lines, circles and ellipses in both 2D and 3D images, [Camurri et al., 2014]; as well as to detect planes, cylinders and spheres, [Drost and Ilic, 2015]. In the HT, the parameters that define a certain primitive are recovered by using a voting scheme. For the initial point(s), all hypothetical primitives are calculated, and single votes are cast for them. This is repeated for all (sample) points, and the hypothetical primitives with the most votes (i.e. that fit to the greatest number of points) are considered to be the ones that fit the data best. The major disadvantage of HT is that the computation cost and memory requirements increase exponentially as the parameter space grows, [Camurri et al., 2014, Drost and Ilic, 2015].

RANSAC - the RANdom SAmple Consensus algorithm was introduced by [Fischler and Bolles, 1981] as a general algorithm to fit a mathematical model to a set of points. It is mostly used for extracting 3D primitives, such as planes, cylinders, cones, spheres and tori. At the beginning, a primitive model is fitted to randomly chosen seed points. The remaining data points are then grouped into inliers and outliers, based on how well are they approximating the primitive shape with a predefined tolerance. After alternating the seed points, a primitive model with the largest set of inliers is chosen. The main benefit of the RANSAC based algorithms is their robustness concerning the large number amounts of noise or outliers, [Schnabel et al., 2007].

Aside from the recognition of primitives from point clouds, there are some approaches on voxel-based models as well. [van der Glas et al., 2002] developed a method for determining the position and size of spheres in unsegmented medical images by using the gradient vector direction. The gradient value for each voxel is mapped on a histogram, and the radius is calculated by finding the maximum values in the histogram. [Cao et al., 2006] proposed a method for detecting circles within individual CT slices. Both methods are Hough transform variations. [Xu et al., 2017, Rizzini et al., 2014] combined point cloud based recognition with a voxel-based data structure in order to obtain better spatial categorisation of the points.

After successful recognition of simple primitives, initial point cloud model representation is usually abandoned. Improved (corrected) models are typically represented by more complex representation, such as NURBS surfaces or meshes, [Berger et al., 2014, Raja and Fernandes, 2007].

2.4.3 Segmentation

Segmentation is connected to shape recognition, but it is a different post-processing procedure. The targets in shape recognition are explicitly defined via exact parametrisation, well-defined model or specific features. Segmentation is, however, a process of separating different regions within a model depending on a single property. Image segmentation results in assigning labels to each element (pixel or voxel) of an image in a way that all the elements with the same labels share the same property. A typical task in medical image processing is segmentation of different organs or body structures based on the restored X-ray attenuation coefficients. The same procedure can be applied to distinguish between different materials (or simply between material and air) after reconstructing a general 3D object.

Since the main goal is 3D replication in a RP environment, the reconstructed voxelised model has to be transformed into a 3D printable form. A standard way is to obtain a B-rep of a voxelised data and then slicing it in order to get layer-wise information. However, a reconstructed voxelised matrix can be comprehended as a stack of layers and does not have to be transformed into B-rep and sliced. By extracting the region boundaries (segmentation) on each layer and filling them accordingly, a layer-wise 3D printable model can be directly obtained. Besides, some AM processes manufacture objects directly from voxels, [Tedia and Williams, 2016].

Segmentation methods can be divided into either manual or automatic, [Brock, 2013]. Manual methods rely massively on user's interaction, who for instance, manually marks the contours of the desired regions. Automatic methods extract the regions either completely autonomously or with minimal user's interaction. Segmentation itself can be achieved by intensity, by region growth or by boundaries. Segmentation by intensity separates different regions according to a user defined threshold. While region growing methods split regions by following the same intensity of a predefined seed in its neighbourhood, boundary orientated methods detect sharp changes in intensities with, for instance, spatial filtering. Segmentation methods are often very specialised for a certain type of applications, [Birkfellner, 2015].

One of the most widely used automatic segmentation methods is intensity-based *thresholding*. Here, all the pixels are binary classified into two groups: pixels with grey levels above the threshold value and pixels with grey levels below the threshold value. Region contours can be determined by tracing the boundary between the neighbouring pixels with different binary values. Another common intensity-based segmentation method is *watersheding*. Here, the grey levels of pixels are viewed as heights. The contours of different regions are then considered to be the ridge lines of a relief. Depending on a defined threshold (height), the corresponding contour can be extracted. The boundary contour resulting from watershading is considerably smoother than the stepwise shaped boundary contour from thresholding.

2.4.4 Mesh simplification

Meshes that result from the 3D digitization of objects typically contain a large amount of densely positioned points. To remove the superfluous level of detail, reduce the storage and speed up the rendering, a 3D mesh can be simplified. Common algorithms for mesh simplification can be classified into incremental decimation, vertex clustering and re-tiling.

Incremental decimation methods, introduced by [Schroeder et al., 1992], rely on an iterative greedy procedure for gradual mesh decimation. Examples of the decimation operations are vertex removal, edge removal or complete mesh element removal, [Shontz and Nistor, 2013]. At the beginning, a mesh is divided into regions. An error is measured for all possible decimation operations in each region. According to the measured error, operations are enqueued (best-come-first). The operations are then performed successively until the maximally allowed regional error is achieved. After each decimation operation, the error is to be recalculated. Incremental decimations (or local simplification strategies in general) are the most common mesh simplification methods, due to user's explicit control over the mesh precision (allowed maximal error) and geometrical attributes (desired number of faces or vertices), [Talton, 2004].

Vertex clustering was initially proposed by [Rossignac and Borrel, 1993] and it can be classified as a global simplification method since it applies a global strategy to the input mesh as a whole, [Talton, 2004]. The basic approach in the representative algorithms is clustering of the vertices based on their topological location. One representative vertex is preserved for each cluster group by merging all the vertices from the same cluster. In the end, the local neighbourhood of the cluster representatives is re-meshed by collapsing topology. Key advantages of this class of methods are their speed-efficiency and implementation-simplicity, [Talton, 2004].

Re-tiling methods, introduced by [Turk, 1992], are based on generating simplified meshes on an entirely new set of vertices. After generating random points on the original mesh surface, repulsion and diffusion are applied to position the points uniformly. The last step of the algorithm is the tessellation of the points.

A direct consequence of the mesh simplification is a geometry modification. There are different error metrics for measuring the geometrical fidelity between two meshes, such as the quadric metric and the Hausdorff metric. In the quadric metric, the global error is computed as a sum of squared distances between vertices of the original mesh and corresponding faces in the simplified mesh, [Garland and Heckbert, 1998, Ronfard and Rossignac, 1996]. Within the Hausdorff metric, however, the global error is defined as a maximum of the two one-sided Hausdorff distances. The one-sided Hausdorff distances are defined as the maximal minimal distance between the original (simplified) vertices and the simplified (original) surfaces. An example of the realisation of an error measuring tool between two meshes is given by [Cignoni et al., 1998].

2.5 Additive Manufacturing

According to ISO/ASTM 52900:2015 norm, Additive Manufacturing is defined as a layer-wise process of adhering materials in order to fabricate parts out of 3D models. As a layer-wise process, it opposes formative and subtractive manufacturing approaches. One of its main characterisations are straightforward usability, fast fabrication and usage of low-cost materials and equipment, [Gebhardt, 2013].

Application fields of AM parts are very versatile and their deployment is according to [Wohlers et al., 2016] divided into:

- functional parts (32.5%)
- prototypes for fit and assembly (16.2%)
- education and research (10.1%)
- visual aids (8.5%)
- patterns for metal castings (8.3%)
- presentation models (8.2%)
- patterns for prototype tooling (7.4%)
- tooling components (7.2%)
- other (1.5%)

In the following section different AM processes will be introduced and compared. Afterwards, the process planning aspects of an extrusion based AM branch will be presented.

2.5.1 Classification

Since 1987 there have been more than one hundred different physical processes for the direct computer-based manufacturing of physical objects in the field of AM. They can be divided into the following groups, [Gebhardt and Hötter, 2016]:

- Sintering / melting (powder melting or melting within a powder bed)
- Three Dimensional Printing (binder injection into powder)
- Fused Layer Modelling (extrusion)
- Stereolithography and Polymer printing (photopolymerisation)
- Layer Laminate Manufacturing (thin film cutting)

To fabricate a 3D model via AM, its decomposition into layers is necessary. This process is called *slicing*, and it is often directly connected to the particular AM tool. Only after the creation of the mathematical layer (additive) information, conversion into machine readable instructions can be performed. In addition to setting up the geometrical data, further issues are to be determined: optimal manufacturing orientation (stability aspects and supporting material minimisation) and material and process specifications.

The standardised industrial input format for the representation of 3D models in RP environment is STL (STereoLithography), [Gebhardt, 2013]. An STL represents a triangulated surface, with a set of vertices and a facet normal constituting each triangle (facet). STL format is prone to faults, which can be classified in faults that arise from non-manifold surfaces, open surfaces and surfaces with inverted normals, [Willis et al., 2007]. These faults lead to non-waterproof models and can have either no, little or major impact on the final parts, depending on the undergoing AM process.

A 3D Object is built in a way that successive layers adhere to each other. Neighbouring layers can be bonded either by glueing (as in laminating processes), sintering or by forming a chemical bond. The latter can develop from solidification of a molten material (such as in extrusion processes) or photo curing (applied on photopolymers), [Kulkarni et al., 2000].

Depending on which chemical bond particular AM process uses to adhere to the layers, different materials come into play. Within the broad range of materials used in AM, one can find plastics (polymers), ceramics, metals, paper and composites. However, primarily used materials are polymers and metals. Regarding polymers, various thermoplastics (such as ABS, PLA, PA and PC) and thermosets (such as epoxy and acrylic) are in use. A wide range of metals is applicable as well, such as steel, titanium, aluminium and platinum, [Wohlers et al., 2016].

Depending on the material and the AM process, the model precision in the printing plane (defined by x- and y-axis) and the precision in the building direction (z-axis) are differently influenced. The model precision in the printing plane is limited for polymerisation and thin film cutting by the plotter, whereas for sintering, extrusion and ballistic printing systems it is limited by the diameter of the printing units. For the AM systems with printing heads, either the printing head itself or the plotter can be the precision limiting factor, [Gebhardt and Hötter, 2016].

Different AM processes determine the features other than material and precision as well. These aspects are (among others) tolerance to the input files, post-processing (surface finishing and removal of the remaining material) and building of the supporting structures for overhanging geometries. For sintering and binder injection into powder bed, the remaining powder serves automatically as a supporting structure, while for other printing systems this is not the case.

2.5.2 Process planing in extrusion-AM

Information presented in this section was published by Prša et al. in July 2015 at the 2015 International Conference on Computer, Information and Telecommunication Systems (CITS), [Prša et al., 2015b].

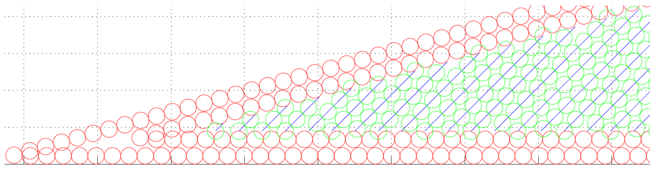
In the Fused Layer Modelling, successive layers are being solidified. Deployed fabrication strategy is the extrusion of a certain amount of nearly-molten material on a building platform, while a pre-calculated path on each layer is being followed. Successive accumulation of material in a layer-wise fashion results in a 3D printed part. The

AM technologies that deploy this approach can be divided into fused deposition modelling (FDM) and droplet generation. In FDM continuous strands are being extruded, while in droplet generating it is the discontinuous droplets that are being extruded.

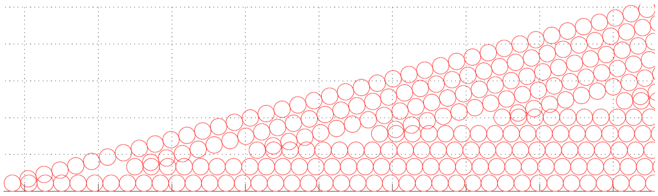
To manufacture a 3D part, as a part of process planning, the corresponding geometry of a CAD model has to be transformed into layer-wised paths, which a machine can easily interpret. The main steps in the process planning are slicing, orientation determination, path (trajectory) planning and generation of the support structures, [Kulkarni et al., 2000].

While slicing itself determines the contours (boundaries) on each layer, path planning determines the strategy for the interior filling. Many authors report about the importance of the path planning process and its impact on the quality of the printed parts, [Kulkarni et al., 2000, Schwaiger et al., 2013, Jin et al., 2013, Agarwala et al., 1996, Qiu et al., 2001, Tarabanis, 2001].

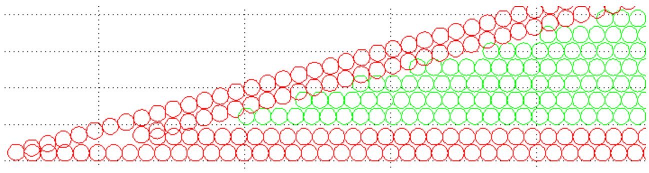
Available filling methods typically include one or more boundaries (contours), which are followed by raster lines (zig-zag hatching), [Kulkarni et al., 2000, Jin et al., 2013, Agarwala et al., 1996, Qiu et al., 2001]. Other proposed filling methods are: repetitive contours offsetting, [Tarabanis, 2001], NURBS-curves offsetting [Tiller and Hanson, 1984], domain decomposition [Bertoldi et al., 1998], spiral curve filling [Kulkarni et al., 2000, Jin et al., 2013] and dense packing [Prša et al., 2013].



(a) Raster filling.



(b) Contour filling.



(c) Dense filling.

Figure 2.6: Droplet-based simulation of three different filling strategies.

Raster filling consists of parallel lines that change the angle with each new layer, typically for 90°, [Nevin and Mehrdad, 2014]. When applying the contour filling, contours are repetitively shifted toward the interior, for as long as possible. If the deposition unit is spherical, dense packing strategy assures that the droplets in the interior are placed as close as possible, following the hexagonal packing. The simulation of the raster, contour and dense filling for droplet generating 3D printers is shown in the Figure 2.6.

2.6 3D Replication

Processes which typically occur in a 3D replication in RP environment are shown in Figure 2.7. The user starts the procedure by object scanning, i.e. obtaining and processing multiple images of an object. These serve as a starting point of a reconstruction, which results in a 3D model. Data collection via scanning and reconstruction are typically parts of an enclosed digitalisation process. After obtaining a digitalised 3D model (point cloud or voxelised data), a separate post-processing unit is necessary in order to adopt the model for its fabrication. Post-processing unit provides preparation steps, such as data conversion, analysis, evaluation and repairing. With that, the user can start an independent RP process.

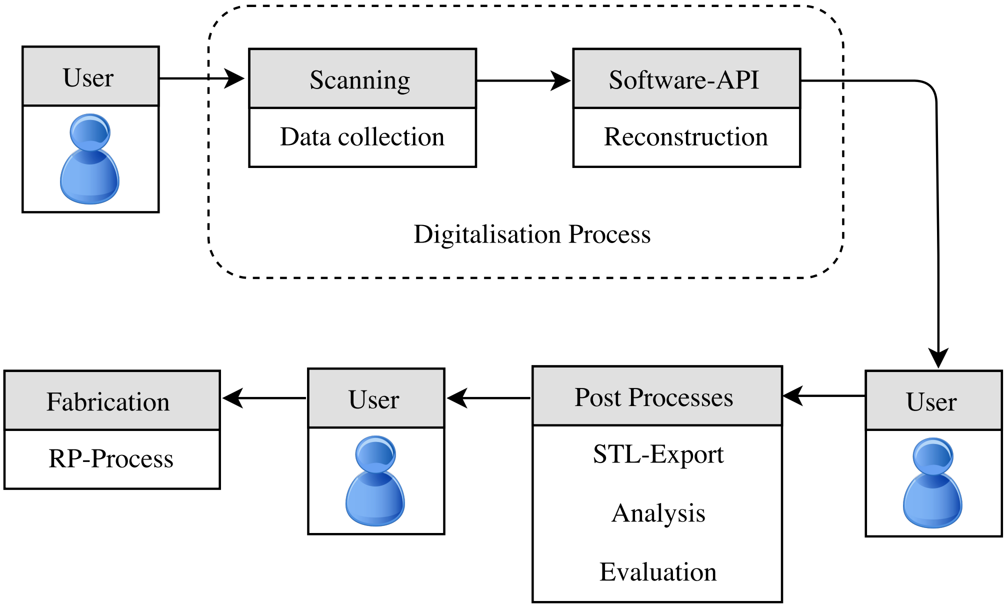


Figure 2.7: State of the art in 3D replication.

2.6.1 State of research

So far AM techniques Fused Layer Modelling (FLM) and stereolithography were combined with either optical or computed tomographical 3D digitization techniques. In [Biglino et al., 2016] CT and stereolithography were combined to obtain cardiovascular 3D models. An evaluation of the fused filament fabrication of a CT dataset was presented in [Bortolotto et al., 2016]. In [Xiong et al., 2008] optical 3D digitization was combined with FLM in maxillofacial prosthetics. [Skala et al., 2013] realised a 3D replication of a coin using a flatbed scanner and AM, whereas 3D copies of dental plasters via FLM were described in [Kasparova et al., 2013]. In [Drosse et al., 2008] for the therapeutic strategies of bone defects, a CT-dataset was processed and used to create 3D models via stereolithography, see Figure 2.8. The integration of the replication processes is at times tightly related to a specific area of application, such as the analysis of bone architecture in osteoporosis, [Borah et al., 2001].

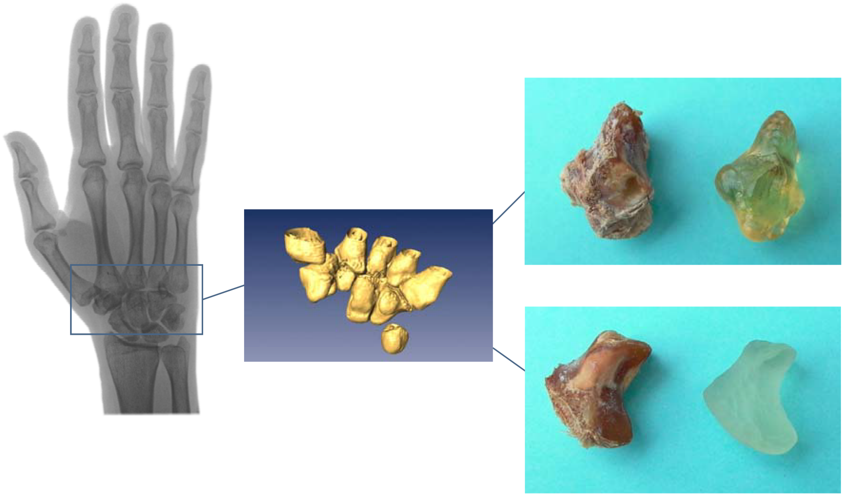


Figure 2.8: Combining CT with stereolithography for the therapeutic strategies of bone defects, taken from [Drosse et al., 2008].

In [Colombo et al., 2006] a novel technique, which combines both optical and computed-tomographical scanning techniques in a prosthesis socket design was introduced: a non-contact laser technique was applied to obtain external geometry (bones geometrical model), while CT and MRI imaging technologies were applied for obtaining internal structure (soft tissues and muscles).

An approach that leads to the direct coupling of optical digitization and AM by automating the data preparation step was proposed by [Liu et al., 2003]. The authors developed automated slicing and segmentation of point clouds (resulting output from optical scanners) in order to generate layer-wise models directly. An improvement of this approach regarding adaptive slicing that yields the constant shape error on each layer was given in [Wu et al., 2004].

2.6.2 Available tools

In the last couple of years first “all-in-one” RP replicating machines have appeared on the market: *Zeus* from AIO Robotics (Figure 2.9a), *Personal Fabricator* from FABtotum, *Da Vinci 1.0 3-in-1* from XYZprinting and *Adam* from Northype. *Zeus* unites printing, scanning, copying and faxing within a single closed housing, while *Personal Fabricator*

(a) Zeus from AIO Robotics, copyright by ©AIO Robotics

(b) MakerBot Replicator 2 with Digitizer, copyright by ©MakerBot

Figure 2.9: Available 3D replication tools.

and *Adam* join printing, scanning, milling, engraving and cutting. *Da Vinci 1.0 3-in-1* enables printing, scanning and laser engravings. All four of these machines have realised a digitizing procedure via optical methods, meaning that only the outer surface of objects can be detected.

Several other 3D-Printing manufacturers enable connections with external 3D Scanners. An example of that approach is *ZMorph* — initially established as a 3D Printer, but with the time evolved into a multi-functional RP system with interchangeable tool heads for milling, scanning and cutting. 3D optical scanners, such as Microsoft *KinectFusion*, MakerBot *Digitizer* (Figure 2.9b), *FabScan* and *FabliTec* are introduced recently to the market as external optical scanners specialised for the data acquisition in RP environment.

2.7 Limitations

2.7.1 Missing replication entity

3D digitization and fabrication are in general not coupled in a single unit, but instead used as *separate and independent processes*, [Bidanda and Geng, 2016]. Additional preparation steps in between are therefore necessary, such as conversion between different model representations, model modification and customisation. These processes typically require several manual steps, which make the connection between 3D digitization and different AM processes indirect and difficult.

When using detached, independent tools for replicating 3D objects, all available knowledge is not fully exploited. For the preceding processes (data acquisition, reconstruction and post-processing), there are specific fabrication requirements and specifications that should be taken into account. These are often already familiar before starting the replication and are simply overlooked. Important model distinctions are with that possibly neglected, while valuable time is spent on irrelevant features.

The currently available means of 3D replication tackle these challenges only partly. Moreover, there are no built-in approaches that can detect the interior of objects since they are limited to the *outer surface detection* due to the utilised optical 3D digitizing techniques.

Both contact methods and non-contact optical methods for 3D digitization are limited on capturing easily accessible outer features. Optical methods have additional constraints concerning coloured, transparent and reflective surfaces. Contact methods are in turn prone to damaging the object (partly or entirely) and sensing probe, require complicated path-planning and result in relatively time-consuming data acquisition, [Raja and Fernandes, 2007].

X-ray, magnetic resonance and ultrasound imaging are the 3D digitization approaches that enable nondestructive inspection of both internal and external structures. However, MRI and ultrasound imaging of solid objects is severely *restricted by the applicable materials*. In MRI, targeted phosphorus atoms are not present in most of the common objects. The signals of hydrogen protons, which are targeted in tissues in medical MRI applications, are harder to detect in general solids and therefore MRI of solid objects is still not very common, [Rinck, 2017]. Ultrasound imaging is strongly limited by the low reflection rate of the ultrasound waves, which is the case for tissues, but not for the conventional solids, [Herment et al., 1987]. Moreover, ultrasound waves are disrupted by air or gas which makes them impractical in general RP environment. In contrast, X-ray imaging is applicable to a wide range of materials and object sizes, [Christoph and Neumann, 2012].

2.7.2 Unsuitability of X-ray apparatuses for RP environment

Cone-beam X-ray apparatuses for 3D digitization are typically realised via an X-ray source and a flat detector, with an underlying cone-beam geometry set-up. As already presented in Section 2.2.4, two commonly used approximative algorithms for 3D cone-beam reconstruction are the SART and the FDK algorithms. SART reconstruction results are superior to FDK if the projection data is noisy, obtained in a limited rotational range or very limited in number, [Tita, 2007]. Moreover, SART, unlike FDK, is insensitive to the small distances between the x-ray source and the detector (which is often the case for desktop X-ray apparatuses). FDK is however advantageous regarding the computational time, [Kalender, 2011]. Since the two algorithms have different strengths and weaknesses, it is desirable to choose different algorithms for various applications.

However, the *reconstruction unit* is usually realised as a “black box” for the users and offers no insight into the underlying processes. Users of general X-ray based devices have thereby no choice concerning the reconstructing strategy and have altogether very limited control over the inner functions, [Yu and Leng, 2016]. Moreover, the reconstruction results are often visible only upon finishing the whole reconstruction calculation, which offers no opportunity for users to get insight into the actual reconstruction state. With this, potential user-driven mistakes, such as the false setting of the parameters, can be changed only after the reconstruction. SART, as an algebraic iterative reconstruction technique, leads to a gradual improvement of the model quality of the reconstructed model. The missing visualisation of the intermediate SART-reconstruction results could also lead to unnecessary long computations.

In order to operate an X-ray apparatus and to set up the measurements, multiple steps are necessary. Setting up of the *relevant X-ray parameters* is typically accomplished manually. It is therefore strongly dependant on a personal judgement and prone to mistakes. Correct decision making upon appropriate X-ray parameters requires technical knowledge and time; hence it is not suitable for user-friendly RP environment. Regarding the setting up of the relevant X-ray parameters in the medical applications, there is a predefined look-up table available which associates X-ray parameters with different body structures. This is possible since the sizes and material structure of different body parts are well known and constant. Due to the high versatility of general objects that can be digitized in industrial applications, no predefined detailed look-up table is existing. Instead, only very rough guidelines for a big group of materials (such as plastic) is familiar, [Christoph and Neumann, 2012].

2.7.3 Limited or missing post-processing

Upon obtaining a raw 3D voxel-based model from an X-ray reconstruction unit, certain preparation and formatting is needed. This includes geometry corrections, the conversion into a suitable B-rep representation (STL), slicing and optimal orientation. While the machine-specific software offers some of the model processing, typical model preparation depends on several external software units and requires skilled users, [Gebhardt, 2013].

Due to the high versatility of existing AM processes, requirements on the input models vary as well. Modular and straightforward *adjustment on different AM tools* from one post-processing unit is however missing. The printing resolution differs significantly for different AM tools. By neglecting this information, valuable fine details could be lost (if model's level of detail is significantly below the AM resolution) or too much unnecessary effort could be invested (if model's level of detail is significantly above the AM resolution). It would be therefore reasonable to adjust the level of detail of printable models. Since the meshes that result from transforming voxel-based models typically contain a large amount of vertices, i.e. high level of detail, modular mesh simplification should be performed. Most known mesh simplification algorithms reduce the overall number (percentage) of vertices or surfaces; or simplify the mesh in such a way that the user's defined error bound controls the overall error metric, [Talton, 2004]. This, however, does not comply with the goal of eliminating the details that are anyway not possible to be distinguished by a particular AM tool of choice. A simplification tool that removes too close vertices while the general shape is being preserved is therefore missing.

During the replication process, model representation is changing several times. Voxel-based models are typically transformed into point clouds, which are in turn transformed into B-reps, such as standardised triangular meshes. Finally, triangular meshes are sliced in order to obtain layer-wise contours and infills that are interpretable by the AM tools. With each of these data conversions, additional errors are accumulated. Creating a B-rep model and then slicing it to obtain layer-wise representation for AM is for voxel-based models arguably redundant. Since the voxel data already contains layers of information, the necessary AM layer-wise representation can be extracted directly from the voxel-data by segmentation. By neglecting the possibility of *direct segmentation* on a voxel-level, valuable time for omittable conversions to point clouds and B-reps is lost, and geometry deviations are possibly increased.

Regarding the *automatic primitive shape recognition*, commercial software tools, such as Geomagic, Rapidform, and Solidworks, have automatic shape recognition features. However, they perform poorly, and user interaction was needed to recover partially recognised or unrecognised primitives, [Chang and Chen, 2011]. Regarding the state of research, the vast majority of methods for automatic primitive shape recognition and correction does not work on voxel-based model representations. While the recognition is mostly done on point cloud representations, the corrected models are typically B-reps. Rare voxel-based methods are limited to the automatic recognition of circles and spheres.

2.7.4 Infill faults in extrusion-AM

The limitations of the filling methods in extrusion-AM were published by Prša et al. in July 2015 at the 2015 International Conference on Computer, Information and Telecommunication Systems (CITS), [Prša et al., 2015b].

The two approaches, contour infill and printing of several contours followed by fast printable raster lines, tend to have under- and over-fills. Moreover, in the case of utilising droplet generating 3D printers the spherical deposition unit is not taken into consideration. Domain decomposition and spiral curve filling are limited to certain contour shapes. The disadvantages of raster, contour and dense fillings will be elaborated in more detail.

Raster filling generates two kinds of possible defects: voids near the contours and overfills. The voids can be classified into two groups: sub-perimeter and inter-road voids, [Agarwala et al., 1996]. The sub-perimeter voids arise along the perimeter (boundary) since two parallel hatchings cannot entirely fill the boundary area (if not accidentally hitting the boundary exactly vertically). The inter-road voids are typically caused due to the mechanical reasons. While trying to overcome the under-fill deficits, a trade off between voids and over-fills is possible. A certain overlap of the boundary and raster lines, and an overlap of raster lines will decrease voids, but the final result could be an undesired material hump.

Contour filling generates three kinds of possible defects: core voids, outer-inner boundary encounter and acute corner void. With successive offsetting of the contours, the interior is being filled. Unless the interior width is exactly an integer multiple of the deposition width, a core void occurs. If the geometry involves holes or pockets, sliced layers will contain inner contours in addition to outer contours. Once repetitively offset, outer contours encounter the inner ones, and due to possible different shapes, voids occur. If the geometry contains sharp corners, under- and over-fills appear. While the problem of filling the sharp corners has been addressed in [Prša et al., 2014a, Prša et al., 2014b], a general solution for detecting and filling of voids has not been proposed.

Dense filling has a constraint in its geometrical positioning of the circles, with their centres always enclosing 60°. Whenever the boundary coincides with the dense grid in a way that the lines are not (nearly) parallel, voids appear. Furthermore, if the segment width is not an integer multiple of the interior width, new gaps appear.

Different software tools offer the possibility to elude some of the above-mentioned defects for a specific filling strategy (usually only one) by setting certain parameters (i.e. density, allowed overlap, starting points), but either not completely successfully or the right parameter settings require a lot of experience with the effects visible only after printing. Moreover, there is no general tool which automatically addresses and corrects the defects after launching arbitrarily fillings.

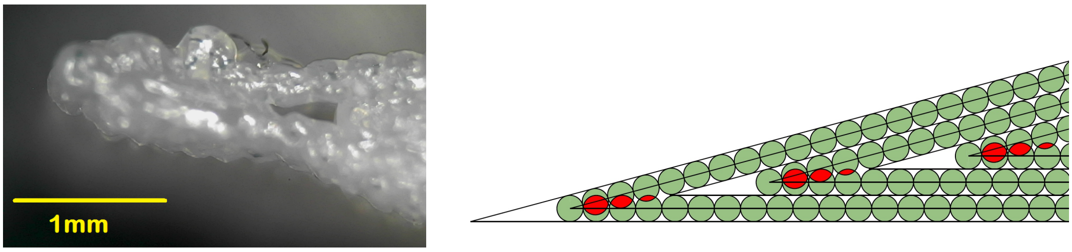
Pointed ends

Limitations of the filling methods for the geometries with pointed ends were published by Prša et al. in November 2014 at the ASME 2014 International Mechanical Engineering Congress & Exposition (IMECE), [Prša et al., 2014a].

The finer the 3D model is, the more points are used to describe a single structure and the probability of presenting ”sudden turns” increases. The sudden turns, once mapped to a 2D slice result in segments, form sharp angles. The angle between the two neighbouring segments influences the level of sharpness of the pointed end.

Prior to printing, the path is shifted from the actual boundary contour for a radius of the ball, so that the printed droplets do not surpass the given boundary. Sharp corners of the pointed ends typically remain improperly filled due to the limiting width of the extruded unit. An example of a 3D printed geometry which contains pointed ends can be seen in Figure 2.10a.

The optimal case (voids and overlaps are minimised) occurs when the length of a printing segment corresponds exactly to the multiple width of the ball (such that there are no voids and no overlapping within a segment between the balls). Therefore, it is assured that one ball is placed exactly in the angle, as far as possible in a way that it does not exceed the boundary lines. Even in this optimal case, the voids and overlapping are occurring. Moreover, the overlapping and voids are repetitive in case of applying the shrinking contour method as the infill strategy, Fig. 2.10b. Infill errors arise independently of the infill strategy (raster or contour filling).



(a) A pointed end printed with a 3D droplet-generating printer: clear problems in representing the tip-like geometry.

(b) A 2D slice with its two segments that enclose the angle of 15°. Coloured in green are the single-filled areas, unfilled areas are white, whereas the double-filled areas are red.

Figure 2.10: Infill faults for the geometries with pointed ends.

3 X-ray based 3D Replication in RP environment

3.1 My approach

The aim of this thesis is to develop a modular 3D replicating concept and its computational realisation. To achieve that, 3D digitization, preparation of the digitized models and fabrication via AM will be coupled. In order to enable nondestructive and precise digitizing of even internal and inaccessible geometries, utilisation of a cone-bean X-ray apparatus is proposed. Its adaptation to a user-friendly RP environment will be made by minimising the unnecessary manual steps, while still enabling insight and control over the reconstruction process. Suitable model preparation and modular adjustment of the data for the fabrication, according to the requirements of the various AM systems, will be conducted upon the reconstruction.

The development is conceptually divided into three blocks, which extend the current disconnected and inflexible replicating process (depicted in Figure 2.7):

User-controlled reconstruction unit

The reconstruction unit is conceptualised as a user-interface which offers multiple possibilities for configuring the reconstruction process. Configuration options include several reconstruction methods from both analytical and iterative classes, settable parameters that influence the reconstruction quality and parameters that define the geometrical cone-beam setting of the reconstruction. A higher level of interaction is enabled by providing the visualisation of the intermediate reconstruction stages. This, in turn, enables more control over the reconstruction process. The reconstruction can be terminated if the desired visual model quality is achieved or aborted at an early stage in case of a faulty parameter set-up.

In order to evaluate arbitrary objects for which real x-ray projections are not available, virtual projections of a digitized model can be created via a cone-beam X-ray simulator. This enables the verification of the pipeline part which is independent of the X-ray projections. After obtaining a reconstructed volume out of simulated projections, the initial and the reconstructed volume can be compared. The corresponding error can then be used as a stopping criterion for the reconstruction.

Automated processing of 3D reconstructed models

Upon completing the reconstruction stage, a raw voxelised model is obtained. The model improvement is performed by the automated detection and correction of planes, which are the most common 3D geometrical primitives. The plane correction is carried out directly at the voxel level, which leads to immediate correction of the errors and avoids their further propagation. Moreover, the direct voxel slicing benefits from the correction at the voxel level. After the geometry correction, automatic model transformation into RP-suitable triangular mesh representation is provided.

Regarding the process planning for droplet-based AM systems, the optimisation of the infill faults is provided. The development of a general software tool automatically addresses and corrects various defects after launching arbitrary fillings.

Effect of the subsequent fabrication process

In order to exploit the knowledge of subsequent AM processes during scanning and post-processing, requirements and specifications of different AM systems are observed and stored within a RP database. Database entries will be taken into account when setting up the scanning parameters in X-ray data acquisition and for post-processing.

In a RP environment, certain materials are more often used, such as polyamide and thermoplastics. For several standard sizes of commonly used RP materials, the voltage and the current of the X-ray tube are beforehand measured, stored in a look-up table and reused when needed. With a gathering of X-ray parameters in a look-up table, data acquisition is simplified and accelerated.

Different precision limitations of AM systems, caused by the various processes, materials and layer thicknesses, are to be taken into account as well. For that, meshes of the reconstructed models are automatically simplified, so the level of detail in a model does not surpass the printing resolution. Database entries contain corresponding AM resolution parameters, tube voltage and current for the utilised material for several standard sizes.

Since the transformation to the standard triangular mesh is avoidable for the 3D replication purposes, direct segmentation (slicing) on a voxel level is provided as well.

3.2 Expected advantages

The approach presented above is the worldwide first automated computational pipeline that enables replicating objects with their external and internal geometries within RP environment. With the direct connection of a 3D digitization and different AM processes that are involved in a 3D replication, the manual usage of several external software units, which require skilled users for the model preparation, is either entirely omitted or automated. Linking of the processes that are involved in a 3D replication takes into account the requirements and specifications of each of the processes. With this, the customisation of the scanning and post-processing processes on different AM procedures, important model distinctions are not neglected and valuable time is spent only on the relevant features.

With the interactive reconstruction unit which enables parameter adjustment, users have more insight and control in the underlying process. The visualisation of the intermediate stages furthermore enables early detection of user-driven mistakes and shortens unnecessary lengthy computations. By offering multiple reconstruction algorithms, each with their own strengths and weaknesses, different aims can be targeted. Versatile applications are thereby supported optimally. Cone-beam X-ray simulator enables the reconstruction evaluation and simulation of arbitrary models, not only of currently standard phantom-based models.

The proposed RP database eliminates the manual setting of the X-ray relevant parameters prior to data acquisition. Gained time, simplicity and independence from the personal judgement, which requires technical knowledge, eases the usage of X-ray apparatuses, and makes them more suitable for the user-friendly RP environment.

Model preparation steps, which include the transformation to the necessary mesh representation, mesh simplification, automatic recognition and correction of planes and direct segmentation, are automated and evoked automatically upon finishing the reconstruction. Mesh simplification leads to the elimination of the models' details that are not possible to be represented by a particular AM process. Storage consumption and time are by that reduced. With automatic plane recognition and detection directly on the voxel-level, unnecessary user interaction and multiple changes of the model representation are avoided. Not only that the accumulated approximation errors are thereby reduced, but also the possibility of handling only the raw voxelised data is given. This enables direct slicing via segmentation, which leads to minimising the number of transformations between different model representations.

A software tool that automatically corrects the filling defects enables model repairing prior to printing. Moreover, deformations are avoided, the stress resistance of the printed parts is improved, and surface roughness is decreased.

All these accomplishments lead to accurate 3D replicas of arbitrarily shaped objects and make the entire replication process more intuitive, faster and modular.

3.3 Distinction

The important issues concerning the X-ray based 3D replication in RP environment that are *not* addressed by this thesis are:

Establishing of a 3D replicating device. The concepts introduced in this thesis *modularly* connect current disconnected processes involved in the replication. A specific solution for a single replicating unit is therefore not given. Establishing of a particular 3D replicating device is accordingly excluded.

Correction of X-ray imaging artefacts. X-ray projections that are used for the reconstructions were not previously improved. The correction of the X-ray imaging artefacts, such as general cone-beam artefacts, beam hardening, cupping and ring artefacts, is not a part of this thesis.

Impact of the orientation. The final results of the scanning and fabricating are somewhat influenced by the orientation of the object/model. The influences that the orientation has on the final replication accuracy are disregarded within the scope of this thesis.

Correction of non-planes. As a part of a post-processing unit, automatic recognition and correction of planes will be presented. Detection and correction of other primitive shapes that are not planes, such as spheres or cylinders, are excluded.

Optimising of the AM data processing. Prior to fabricating a 3D part, corresponding geometrical information has to be layer-wise decomposed and transferred into machine-interpretable code. Even though the aspects of the AM data processing will be addressed, their improvements will not.

Improvement of the voids imposed by droplets' spherical shape. An optimisation of the infill faults will be introduced as a part of the process planning for droplet-based AM systems. In case of extruding spherically shaped droplets along a segment line or a plane, inevitable voids occur, since spheres have only one contact-point with a plane. Improvements of these voids will not be presented here.

4 Concept

Based on the approach presented in Section 3.1, this chapter describes the concept of the computational development of X-ray based 3D replication in a RP environment. A general concept for coupling of X-ray based digitization and AM will be presented in Section 4.1. Conceptual aspects of the corresponding components concerning X-ray digitizing, model processing and AM follow in Sections 4.2, 4.3 and 4.4 respectively.

4.1 Coupling of X-ray based digitization and AM

This section introduces a general concept for X-ray based 3D replication in a RP environment. Key characteristics of RP are simple usability, fast manufacturing, low costs and acceptance of manual steps, [Gebhardt, 2013]. Based on these RP specifications, X-ray based digitization is modularly coupled with different AM systems.

The current disjoint replication procedure was depicted in Figure 2.7. Figure 4.1 shows the main components of the enhanced replication. It is enhanced with the additional interdependencies: increased user control in the X-ray digitizing process (depicted by blue arrows), back coupling with the subsequent fabrication process (dotted red arrow) and automatic adjustment of several processes (green dashed arrows).

The X-ray digitizing unit is enhanced with an interface which enables more user interaction and customisation. Among variable parameters are a reconstruction algorithm (SART or FDK), parameters that influence the reconstruction quality (voxel size, the number of projections and iterations) and parameters that define the geometrical cone-beam setting (the distance between source and detector, the distance between source and object and reconstruction space). The voxel size can be automatically adjusted so that it conforms with the layer thickness of the subsequent AM process. A feedback unit provides intermediate reconstruction results via a real-time visualisation. With this intermediate information, a user has the choice to finish the reconstruction procedure if the results are already satisfactory, adjust the parameters in case of an incorrect setting or simply let the reconstruction to run as it is. In the case of not having access to an X-ray apparatus, the developed X-ray simulator enables the creation of the virtual projections. These are then further treated the same as the regular X-ray projections. However, the original model out of which virtual projections are simulated enables a direct comparison with the intermediate versions of the reconstructed models. These intermediate errors can be alternatively used as an automated stopping criterion in the reconstruction procedure.

Upon the reconstruction, the resulting voxel-based model is corrected by an automated unit for plane detection and correction. All corrections are performed directly on the voxelised model. Since the AM systems, in general, cannot handle the voxel-based data,

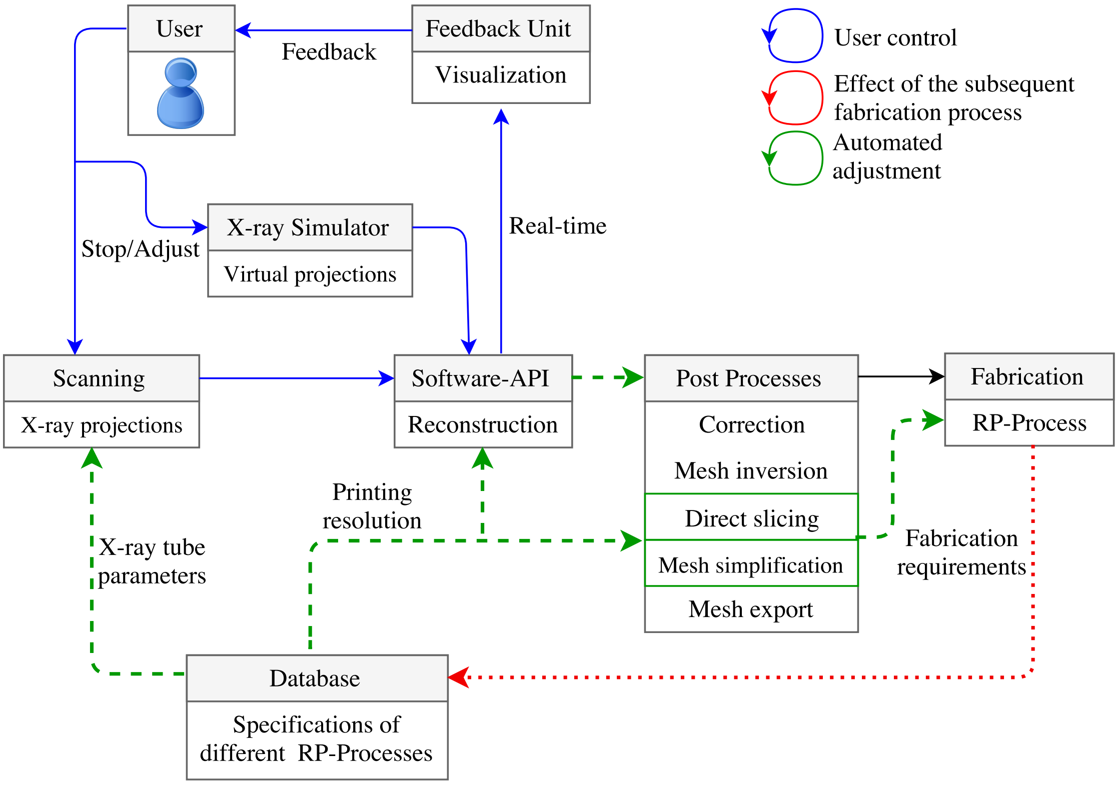


Figure 4.1: Framework for X-ray based 3D replication in RP environment with three interdependencies: user control in the X-ray digitizing process (blue arrows), back coupling with the subsequent fabrication process (red dotted arrow) and automatic adjustment of linking processes (green dashed arrows).

a conversion to either sliced layers or triangular meshes is necessary. Both of these conversions are supported. With the direct slicing through segmentation previously corrected models are translated into an AM machine readable code. The mesh export function enables the transformation from the voxel-based model into a (standardised) triangular mesh. Based on the printing resolution of an AM system of choice, a mesh is automatically simplified in such a way that no mesh vertices are closer than the printer resolution. With all these functionalities, previous post-processing is enhanced, automated and adopted on the AM application field.

The effects that the fabrication processes have on both scanning and post-processing procedures are extracted in a form of fabrication specifications. These, in turn, affect the database in the form of previously gathered database entries. Each AM process has a limited number of materials, which, together with the printing technique, limit the printing precision. Therefore, database entries link different AM processes and materials with the printing precision. Database entries additionally link common materials and object sizes to the optimal X-ray tube parameters. This structure enables the automated setting of X-ray tube parameters, automated mesh simplification and direct slicing.

4.2 X-ray based digitization

4.2.1 Cone-beam reconstruction unit

The main steps of the reconstruction unit are:

1. preprocessing of the projection data
2. object reconstruction via SART or FDK
3. model refining

The starting point of the reconstruction unit is the collection of previously obtained projections and corresponding empty projections. In the preprocessing phase, these are transformed according to the Equations (2.1), (2.2) and (2.3). The reconstruction itself is possible via the iterative SART algorithm or the analytic FDK algorithm. Visualisation feedback is given after processing each of the projection angles or after each iteration. Upon obtaining a reconstructed 3D model, refining with a median filter is conducted in order to reduce the salt-and-pepper noise. The median filter sets each voxel to the median value of its neighbourhood.

SART

The computational steps of the SART reconstruction are described in the Section 2.2.4. Two main functions in this algorithm are `XrayProjection` and `XrayBackProjection`. In the former one, forward (virtual) projections of a voxel-based volume are calculated according to a predefined cone-beam geometry. In the latter one, a volume is calculated by back projecting the difference between the calculated and the real projections.

Prior to computations, a geometry set-up has to be defined. This includes the distance from an X-ray source to a detector dis_{SD} and distance from the source to the centre of the volume/object dis_{SO}. The coordinate system is defined in a way that the z axis represents the rotational axis and the volumetric slices are defined by y-coordinate y, i.e. they are laying in the xz planes. The geometry set-up is depicted in the Figure 4.2.

Projections $\boldsymbol{Proj_{\alpha}}$ are obtained stepwise for each slice by accumulating the volumetric values $\boldsymbol{\mathcal{V}}$ along the rays. In order to do that, for each ray that is associated with the detector coordinates (u, v), corresponding coordinates (u_p, v_p) within each slice have to be found. From the intercept theorem (see Figure 4.2) it is clear that:

$$\begin{aligned} u_p &= \frac{u(dis_{SO} + y)}{dis_{SD}} \\ v_p &= \frac{v(dis_{SO} + y)}{dis_{SD}} \end{aligned} \tag{4.1}$$

In the end, accumulated projection values are scaled due to the different path lengths that each ray traverses. `XrayProjection` algorithm can be summarised as presented in the Algorithm 1.

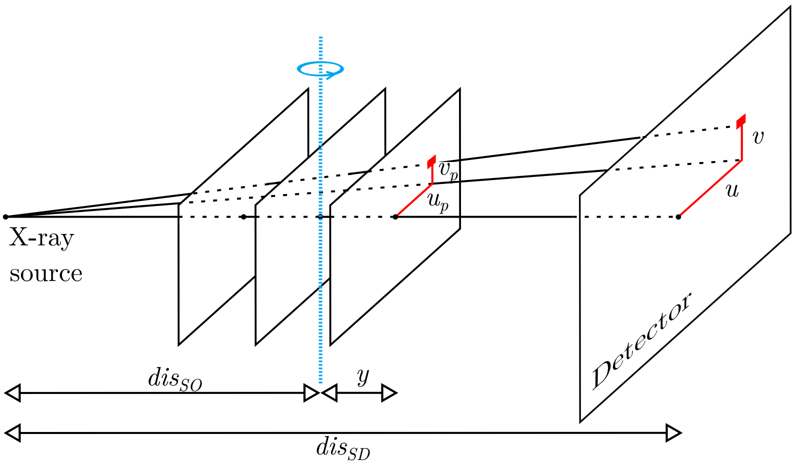


Figure 4.2: Geometry setting with the relations between X-ray source, detector and volume slices.

Algorithm 1 `XrayProjection`

```
Input:
    $\boldsymbol{\mathcal{V}}$ – 3D array of reals
    $\boldsymbol{angles}$ – vector of rotational angles
    $dis_{SO}, dis_{SD}$ – real numbers
    $\boldsymbol{Dpoints}$ – 2D array of detector points
Output:
    $\boldsymbol{Proj_{\alpha}}$ – matrices of reals for every $\alpha$ in $\boldsymbol{angles}$

$\boldsymbol{x}, \boldsymbol{y} \leftarrow$ vectors of coordinates for each z-column of $\boldsymbol{\mathcal{V}}$
$\boldsymbol{u}, \boldsymbol{v} \leftarrow$ vectors of coordinates of $\boldsymbol{Dpoints}$
for all $\alpha$ in $\boldsymbol{angles}$ do
    $\boldsymbol{rx}, \boldsymbol{ry} \leftarrow$ rotated $\boldsymbol{x}, \boldsymbol{y}$ around z-axis for $\alpha$
    $\boldsymbol{\mathcal{R}} \leftarrow$ interpolated $\boldsymbol{\mathcal{V}}$ at $\boldsymbol{rx}, \boldsymbol{ry}$ (rotated $\boldsymbol{\mathcal{V}}$)
    $\boldsymbol{Proj_{\alpha}} \leftarrow \boldsymbol{0}$-matrix, parametrised through $\boldsymbol{u}, \boldsymbol{v}$
    for each xz-plane of $\boldsymbol{\mathcal{V}}$ defined by y-coordinate $y$ do
        $\boldsymbol{u_p} \leftarrow \frac{\boldsymbol{u} \cdot (dis_{SO} + y)}{dis_{SD}}$
        $\boldsymbol{v_p} \leftarrow \frac{\boldsymbol{v} \cdot (dis_{SO} + y)}{dis_{SD}}$
        $\boldsymbol{Slice\mathcal{R}} \leftarrow$ 2D slice of $\boldsymbol{\mathcal{R}}$ at $y$
        $\boldsymbol{SliceProj} \leftarrow$ interpolated $\boldsymbol{Slice\mathcal{R}}$ at $\boldsymbol{u_p}, \boldsymbol{v_p}$
        $\boldsymbol{Proj_{\alpha}} \leftarrow (\boldsymbol{Proj_{\alpha}} + \boldsymbol{SliceProj})$
    $\boldsymbol{Proj_{\alpha}} \leftarrow \boldsymbol{Proj_{\alpha}} \cdot \frac{\sqrt{\boldsymbol{u^2} + \boldsymbol{v^2} + dis_{SD}^2}}{dis_{SD}}$ (componentwise)
```

Upon calculating projections out of the current volume, the difference between the real and the calculated projections (correction $\boldsymbol{Corr}$) is to be back projected. This is done by accumulating the back-projected values for all rotational angles α. For rotated volumetric coordinates corresponding weights w and detector coordinates (u, v) have to be calculated. The latter is done by reversing Equation 4.1. The smearing of the correction values is lastly done along the volumetric slices that are defined by a z-coordinate z, i.e. they are positioned in the xy planes. With that, `XrayBackProjection` algorithm can be summarised as presented in the Algorithm 2.

Algorithm 2 `XrayBackProjection`

```
Input:
    $\boldsymbol{angles}$ – vector of rotational angles
    $\boldsymbol{Corr_\alpha}$ – matrices of reals for every $\alpha$ in $\boldsymbol{angles}$
    $dis_{SO}, dis_{SD}$ – real numbers
    desired output dimensions
Output:
    $\boldsymbol{\mathcal{V}}$ – 3D array of reals

$\boldsymbol{\mathcal{V}} \leftarrow$ 3D array of output dimensions initialised with zeros
$\boldsymbol{x}, \boldsymbol{y} \leftarrow$ vectors of coordinates for each z-column of $\boldsymbol{\mathcal{V}}$
for all $\alpha$ in $\boldsymbol{angles}$ do
    $\boldsymbol{\mathcal{V}}_\alpha \leftarrow$ 3D array of output dimensions initialised with zeros
    $\boldsymbol{rx}, \boldsymbol{ry} \leftarrow$ rotated $\boldsymbol{x}, \boldsymbol{y}$ around z-axis for -$\alpha$
    $\boldsymbol{u} \leftarrow \frac{\boldsymbol{rx} \cdot dis_{SD}}{dis_{SO} + \boldsymbol{ry}}$ (componentwise)
    $\boldsymbol{w} \leftarrow \frac{dis_{SO}^2}{(dis_{SO} - \boldsymbol{ry})^2}$ (componentwise)
    for each xy-plane of $\boldsymbol{\mathcal{V}}$ defined by z-coordinate $z$ do
        $\boldsymbol{v} \leftarrow \frac{z \cdot dis_{SD}}{dis_{SO} + \boldsymbol{ry}}$ (componentwise)
        $\boldsymbol{Slice\mathcal{V}} \leftarrow$ interpolated $\boldsymbol{Corr_\alpha}$ at $\boldsymbol{u}, \boldsymbol{v}$
        $\boldsymbol{\mathcal{V}}_\alpha$ at slice $z \leftarrow (\boldsymbol{Slice\mathcal{V}} \cdot w)$ (componentwise)
    $\boldsymbol{\mathcal{V}} \leftarrow (\boldsymbol{\mathcal{V}} + \boldsymbol{\mathcal{V}}_\alpha)$
```

By computing projection and correction as presented here, values for all the pixels of a detector and all voxels within one slice can be calculated at once.

FDK

The computational steps of the FDK reconstruction are described in the Section 2.2.4. Two main functions in this algorithm are `XrayFiltering` and `XrayBackProjection`. In the former one, a ramp filter is created (bounded absolute value) and convolved with the real projections in the Fourier space. Due to the different path lengths that rays may have, scaling of the projections is performed. Finally, the filtered projections are back transformed from the Fourier space. The `XrayFiltering` steps are presented in the Algorithm 3. In `XrayBackProjection`, a volume is calculated by back projecting filtered projections, as described in the Algorithm 2.

Algorithm 3 `XrayFiltering`

```
Input:
    $\boldsymbol{angles}$ – vector of rotational angles
    $\boldsymbol{Proj_\alpha}$ – matrices of reals for every $\alpha$ in $\boldsymbol{angles}$
    $dis_{SO}, dis_{SD}$ – real numbers
    $\boldsymbol{Dpoints}$ – array of detector points
Output:
    $\boldsymbol{ProjFilt_\alpha}$ – matrices of reals for every $\alpha$ in $\boldsymbol{angles}$

$\boldsymbol{filt} \leftarrow$ ramp filter
$\boldsymbol{FILT} \leftarrow FFT(\boldsymbol{filt})$
$\boldsymbol{u}, \boldsymbol{v} \leftarrow$ vectors of coordinates of $\boldsymbol{Dpoints}$
for all $\alpha$ in $\boldsymbol{angles}$ do
    $\boldsymbol{Proj_\alpha} \leftarrow \boldsymbol{Proj_\alpha} \cdot \frac{dis_{SD}}{\sqrt{\boldsymbol{u}^2+\boldsymbol{v}^2+dis_{SD}^2}}$ (componentwise)
    $\boldsymbol{PROJ_\alpha} \leftarrow FFT(\boldsymbol{Proj_\alpha})$
    $\boldsymbol{ProjFilt_\alpha} \leftarrow IFFT(\boldsymbol{PROJ_\alpha} \cdot \boldsymbol{FILT})$ (componentwise)
```

4.2.2 X-ray simulator

An X-ray simulator enables the reconstruction for 3D models for which the real X-ray projections are not available. Model representations that are supported are triangular meshes and voxelised models. In the case of mesh representation, a model needs to be voxelised. This is done by a ray intersection method, in which the intersections are calculated between many parallel rays and triangular facets. Interior voxels are the ones located between each odd and next even number of intersections, while the rest of the voxels is considered to be exterior. An important step prior to calculating virtual X-ray projections is ensuring that the voxelised model has a sufficient number of exterior voxels so that the rotation does not cut off the bordering areas. Virtual projections are then obtained as described in the Algorithm 1. To mimic a real-world scenario, one can introduce noise to otherwise possibly too perfect projections. This is done by adding random numbers from the Poisson distribution to the projections. After obtaining a reconstructed volume out of simulated projections, one can compare the initial and the reconstructed volume by calculating the difference between each pair of voxel values.

The main steps (see Figure 4.3) of the reconstruction via X-ray simulator are:

1. Voxelisation of the 3D model.
2. Adding exterior voxels on the border.
3. Performing `XrayProjection`.
4. Adding noise to the projections.
5. Object reconstruction via SART or FDK.
6. Model refining.

7. Error estimation.

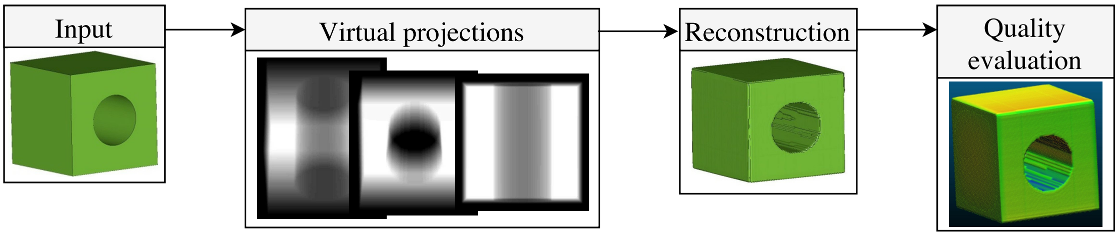


Figure 4.3: X-ray simulator procedure.

4.2.3 Automated setting of X-ray tube parameters

Prior to acquisition of the projections, X-ray relevant parameters have to be set, as described in the Section 2.3.2. These parameters are the current I of the X-ray tube and acceleration voltage V of the X-ray tube. For the each new object, I and V are to be set properly according to the object's size and material. However, the sizes of scanned objects are strongly limited by the dimensions of the closed housing within X-ray apparatuses. For desktop devices, the dimensional limitations are typically up to 100 mm in height (along the rotational axis) and up to 100 mm in width diameter. Within this relatively narrow range of sizes, several standard sizes can be considered. In addition to predetermined sizes, the broad spectrum of all possible materials can be narrowed too. As discussed in the Section 2.5.1, the materials that are typically used in AM are confined. With that, a look-up table which connects commonly used materials and standard object sizes with suitable X-ray tube parameters can be obtained. After specifying the object size and choosing the AM process, manual setting of the X-ray tube parameters is omitted due to the automated fetching of the pre-calculated parameters.

Look-up table

For each material and size, parameters V and I are obtained. This is done in a two-part iterative procedure. In the first part, lower and upper limits of I are iteratively searched for each of the discrete V-steps, in order to obtain a satisfying reconstruction result with the underlying projections. In the second part, lower and upper limits of V are iteratively searched for each of the discrete I-steps.

Projections are considered to be satisfying if there are no parts that are underexposed or overexposed, and if the contrast is strong. Underexposure is an indication of too low tube parameters. The detector in that case does not receive enough X-ray photons, or the photons have a too low energy level. Overexposure indicates too many or too energetic photons, which penetrate the object without being attenuated enough. Both underexposure and overexposure lead to a deficient image contrast of the projections, which in turn leads to reconstructed models that do not coincide with the reality. To avoid such faulty scans, X-ray tube parameters should be chosen so that the resulting image histograms fulfil the following property: the majority of the grey values in the histogram should be within 20 − 80% range of all possible grey values. Figure 4.5 shows the relation between correctly (wrongly) set X-ray tube parameters and corresponding projection radiographs, histograms and resulting reconstructed models. The employed original reference object is depicted in Figure 4.4. There are typically two peaks in a histogram: one for the grey values that are associated with the background and the other one for the grey values that are associated with the object. Isovalue that separates object elements from the background elements is commonly chosen as a grey value in the middle of the two peaks.

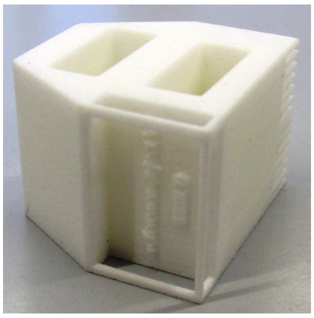


Figure 4.4: Original reference object.

Acceptable parameter pairs (V, I) are typically not unique. In order to provide a single pair in a look-up table, an average V with corresponding average I is considered. However, the voltage of the X-ray tube in desktop X-ray apparatuses is often limited with an upper bound. Therefore, acceptable alternative pairs for lower voltage values are provided as well. This concept is depicted in the look-up Table 4.1.

Table 4.1: Look-up table which connects commonly used materials and standard object sizes with suitable X-ray tube parameters.

Object specifications	Single parameters	Alternative parameters
Material A, Size 1	$(V, I)_{A1}$	$(V, I)_{A1,1}$ $(V, I)_{A1,2}$ $(V, I)_{A1,3}$
Material A, Size 2	$(V, I)_{A2}$	$(V, I)_{A2,1}$ $(V, I)_{A2,2}$ $(V, I)_{A2,3}$
Material A, Size 3	$(V, I)_{A3}$	$(V, I)_{A3,1}$ $(V, I)_{A3,2}$ $(V, I)_{A3,3}$
Material B, Size 1	$(V, I)_{B1}$	$(V, I)_{B1,1}$ $(V, I)_{B1,2}$ $(V, I)_{B1,3}$
Material B, Size 2	$(V, I)_{B2}$	$(V, I)_{B2,1}$ $(V, I)_{B2,2}$ $(V, I)_{B2,3}$
$\vdots$	$\vdots$	$\vdots$

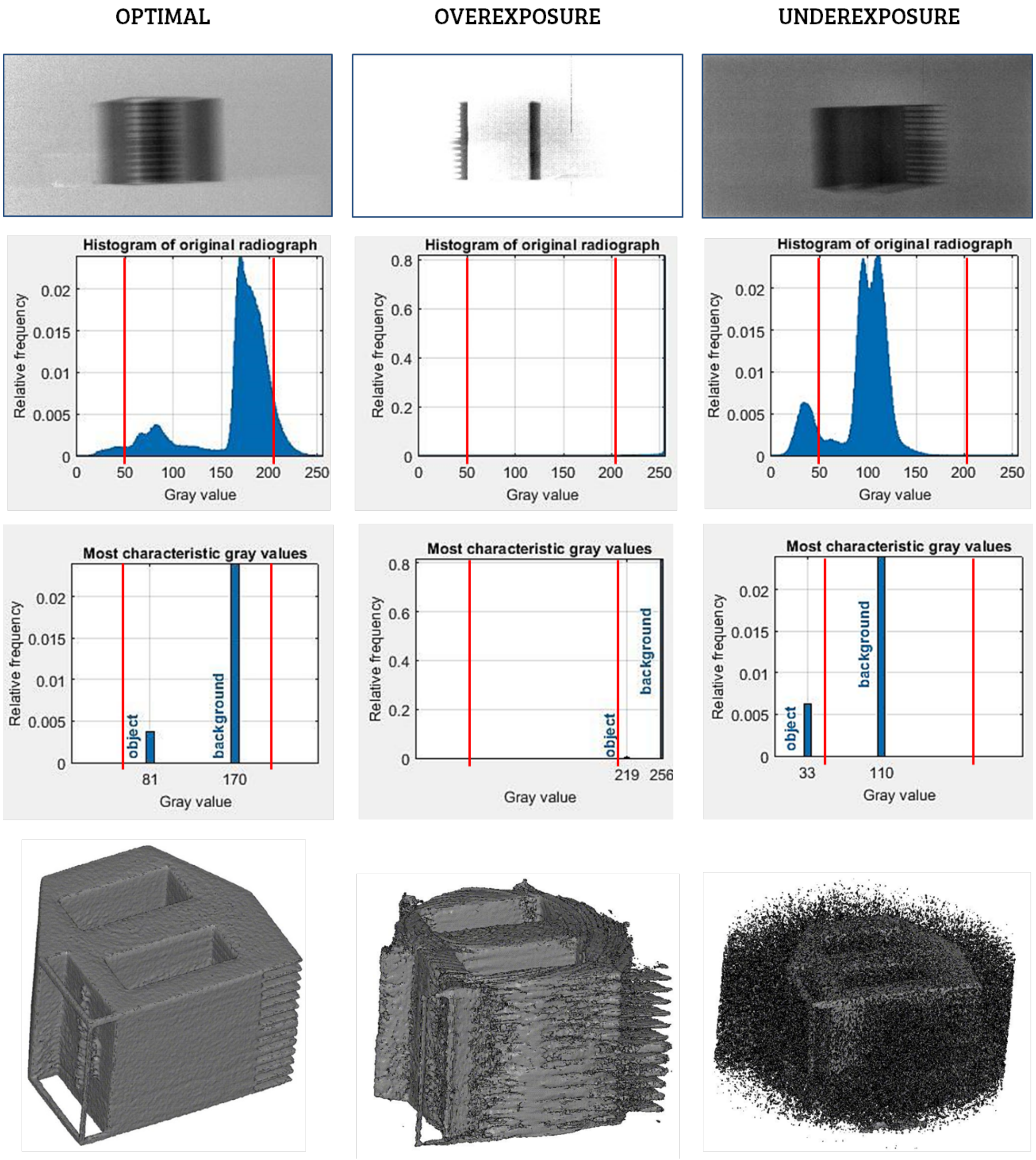


Figure 4.5: The impact of X-ray tube parameters on the reconstructed models: correctly set parameters (left), too high values that result in overexposured radiographs (middle) and too low values that result in underexposured radiographs (right). Projection radiographs are in the first row, corresponding histograms in the second (red lines mark 20% and 80% borders), most characteristic grey values are depicted in the third row, and the resulting reconstructed models are in the bottom row.

4.3 Post-processing

4.3.1 Recognition and correction of planes

The model enhancing process is divided into two steps: plane detection and plane recognition, see Figure 4.6. While plane detection is achieved on a point cloud level, correction is realised on both point cloud and voxel levels.

The starting point of a plane detection unit is a raw voxelised model $\boldsymbol{\mathcal{V}}$. Using the Marching cubes algorithm (introduced in Section 2.4.1), $\boldsymbol{\mathcal{V}}$ is converted into a corresponding mesh, which is defined by a set of vertices *ver* and faces *fac*. After forming a point cloud out of *ver*, planes are detected via RANSAC algorithm (described in Section 2.4.2). Each *plane* is mathematically modelled with four parameters a, b, c and d. Points P that lie on the plane fulfil plane equation: $ax_P + by_P + cz_P + d = 0$. A normal of the plane shows in the direction toward the exterior of the model.

The next step is assigning points to planes by measuring distances between each of the points with coordinates (x, y, z) in *ver* and each of the detected planes. The signed distance *dist* is calculated with:

$$dist := \frac{ax + by + cz + d}{\sqrt{a^2 + b^2 + c^2}}.$$

A point is considered to be associated with a plane if the absolute value of the calculated distance is smaller than the predefined threshold value t. For each point, there exists a (possibly empty) set of planes to which it is assigned, i.e. points can belong to multiple planes. *planePts* indicates a set of all points that are assigned to one plane.

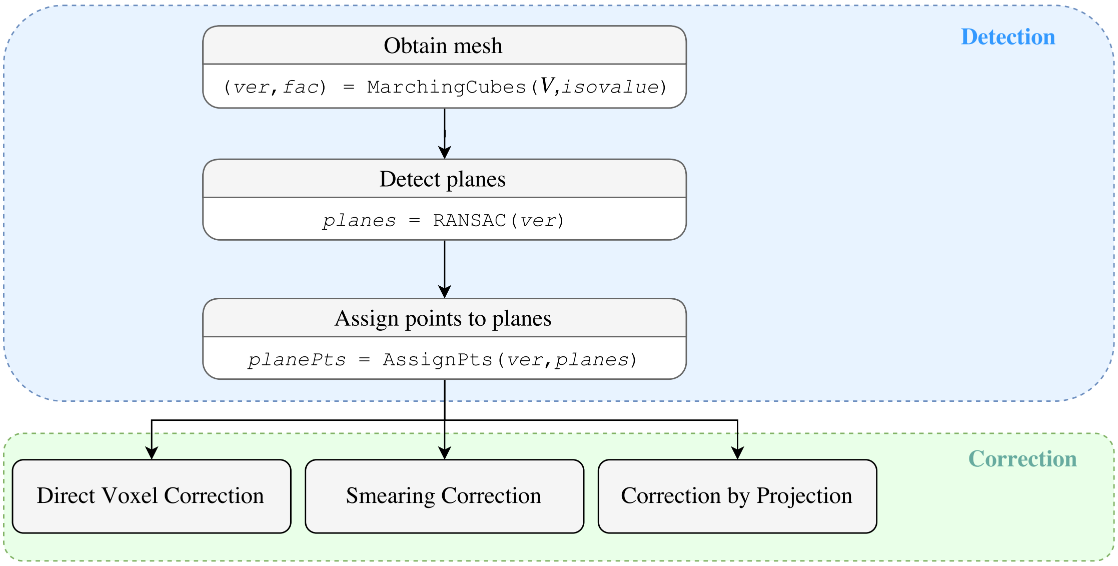


Figure 4.6: Main steps of the unit for plane detection and three correction approaches.

Regarding the correction unit, three approaches are proposed: *direct voxel correction*, *smearing correction* and *correction by projection*. The first two approaches apply correction directly to the voxels, while the latter approach applies corrections to the extracted point cloud. All three methods are implemented and their results will be analysed and compared.

Direct Voxel Correction

The main parts of the `DirectVoxelCorrection` algorithm are the detection of the voxels that are associated with the detected planes and their direct correction. The first step is a calculation of the lower and upper bounds of *planePts* of each detected plane for all three coordinates. The main loop then traverses through all voxels and planes and calculates the corresponding distances. If a voxel is close enough (controlled with a factor t) and within the plane bounds, its distance to the nearest neighbour (NN) among *planePts* will be calculated, as described in [Muja and Lowe, 2009]. Only the voxels that have a close match are considered to be the ones that are associated with the plane. The last step (NN-distance) is necessary since planes may have holes or might have in general non-quadratic-shaped borders and voxels should be assigned only to the plane parts which actually form the model's border. The two checks prior to NN-distance-check are semantically superfluous, but computationally relevant since the NN-distance computation is comparably more expensive.

Algorithm 4 `DirectVoxelCorrection`

```
Input:
    $\boldsymbol{\mathcal{V}}$ – 3D array of reals
    $planePts_i$ – set of all points assigned to every plane $i$
    $a_i, b_i, c_i, d_i$ – coefficients assigned to every plane $i$
    $t$ – real number
Output:
    $\boldsymbol{\mathcal{V}}$ – corrected 3D array of reals

for all planes $i$ do
    $\boldsymbol{lim_i} \leftarrow$ vector of min $x, y, z$ coordinates of $planePts_i$
    $\boldsymbol{Lim_i} \leftarrow$ vector of max $x, y, z$ coordinates of $planePts_i$
for all voxels $V = (x_V, y_V, z_V)$ of $\boldsymbol{\mathcal{V}}$ do
    for all planes $i$ do
        $dist_i \leftarrow \frac{a_i x_V + b_i y_V + c_i z_V + d_i}{\sqrt{a_i^2 + b_i^2 + c_i^2}}$
        if $|dist_i| < t$ & $V \in [\boldsymbol{lim_i}, \boldsymbol{Lim_i}]$ then
            $dist_{NN} \leftarrow NNdistance(V, planePts_i)$
            if $dist_{NN} < t$ then
                if $dist_i > 0$ then
                    $\boldsymbol{\mathcal{V}}(V) \leftarrow$ reduced $\boldsymbol{\mathcal{V}}(V)$
                else
                    $\boldsymbol{\mathcal{V}}(V) \leftarrow$ increased $\boldsymbol{\mathcal{V}}(V)$
```

After detecting the voxels that are associated with a plane, their correction is carried out. If a voxel is located off the plane in the direction of a plane normal (outside the model), its attenuation value is reduced. However, if the voxel is located off the plane in the opposite direction of a plane normal (inside the model), its attenuation value is increased (the implementation details follow in Section 5.2). With this, voxels that are outside the model are approaching the attenuation of air (0), and the voxels that are inside the model are approaching the attenuation of the object's material (1). Algorithmic steps of the `DirectVoxelCorrection` approach are presented in the Algorithm 4.

Smearing Correction

The main parts of the `SmearingCorrection` algorithm are the detection of the voxels that are close to the detected planes, calculation of the next neighbouring voxels along the plane normals and their correction by smearing of the attenuation values.

As in the `DirectVoxelCorrection`, the lower and upper bounds of each detected plane are calculated at the beginning. The main loop includes several iterations through all the planes and starts with the calculation of the indices for the next neighbouring voxels (*getNextVoxel*). Voxels that are outside of the model are being smeared towards the interior (opposite normal direction), while the interior voxels are being smeared towards the exterior (normal direction). Indices $\boldsymbol{Indx}$ are calculated by adding a normalised plane normal $\boldsymbol{n}$ of the corresponding orientation to the actual smearing end-position $\boldsymbol{\Delta}$ within a voxel. At the beginning, all $\boldsymbol{\Delta}$-s are initialised on central voxel positions. With each iteration, $\boldsymbol{\Delta}$-s are updated in a way that they indicate the distance from the centre within the current voxels.

While traversing through all voxels and planes, corresponding distances *dist* are calculated. If a voxel is close enough and within the plane bounds, the attenuation value of its next neighbour is set to a linear combination of its attenuation value and the current attenuation value of its next neighbour. Before entering a new iteration cycle, $\boldsymbol{\Delta}$-s are updated by calling *getNext*Δ. This is achieved by adding normalised $\boldsymbol{n}$ to the actual $\boldsymbol{\Delta}$ positions and retrieving the distances from the centre within new voxels. The two checks prior to correction are again semantically superfluous but speed up the computation by correcting only relevant voxels. With this, the voxels that are outside the model are smearing low attenuation values toward the borders, and the voxels that are inside the model are smearing high attenuation values from the other side of the same borders. Sharp edges are thus expected to arise. Algorithmic steps of the `SmearingCorrection` approach are presented in the Algorithm 5.

Correction by Projection

In the `CorrectionByProjection` approach voxel representation is completely abandoned, and all the corrections are carried out on the previously extracted *planePts*. These points are simply orthogonally projected onto the planes. This is accomplished by calculating the distance between each of the points in *planePts* and corresponding planes, multiplying the distance with the unit normal plane vector $\boldsymbol{n}$ and subtracting

Algorithm 5 `SmearingCorrection`

```
Input:
    $\boldsymbol{\mathcal{V}}$ – 3D array of reals
    $planePts_i$ – set of all points assigned to every plane $i$
    $a_i, b_i, c_i, d_i$ – coefficients assigned to every plane $i$
    $t$ – real number
Output:
    $\boldsymbol{\mathcal{V}}$ – corrected 3D array of reals

for all planes $i$ do
    $\boldsymbol{lim_i} \leftarrow$ vector of min $x, y, z$ coordinates of $planePts_i$
    $\boldsymbol{Lim_i} \leftarrow$ vector of max $x, y, z$ coordinates of $planePts_i$
    $\boldsymbol{\Delta_i^{In}} \leftarrow (0, 0, 0)$
    $\boldsymbol{\Delta_i^{Out}} \leftarrow (0, 0, 0)$
    $\boldsymbol{n_i} \leftarrow (a_i, b_i, c_i)$
for several iterations do
    for all planes $i$ do
        $\boldsymbol{Indx_i^{In}} \leftarrow getNextVoxel(\boldsymbol{n_i}, \boldsymbol{\Delta_i^{In}})$
        $\boldsymbol{Indx_i^{Out}} \leftarrow getNextVoxel(-\boldsymbol{n_i}, \boldsymbol{\Delta_i^{Out}})$
    for all voxels $V = (x_V, y_V, z_V)$ of $\boldsymbol{\mathcal{V}}$ do
        for all planes $i$ do
            $dist_i \leftarrow \frac{a_i x_V + b_i y_V + c_i z_V + d_i}{\sqrt{a_i^2 + b_i^2 + c_i^2}}$
            if $|dist_i| < t \;\&\; V \in [\boldsymbol{lim_i}, \boldsymbol{Lim_i}]$ then
                if $dist_i > 0$ then
                    $\boldsymbol{\mathcal{V}}(V + \boldsymbol{Indx_i^{Out}}) \leftarrow$ lin.comb.of $\boldsymbol{\mathcal{V}}(V)$ and $\boldsymbol{\mathcal{V}}(V + \boldsymbol{Indx_i^{Out}})$
                else
                    $\boldsymbol{\mathcal{V}}(V + \boldsymbol{Indx_i^{In}}) \leftarrow$ lin.comb.of $\boldsymbol{\mathcal{V}}(V)$ and $\boldsymbol{\mathcal{V}}(V + \boldsymbol{Indx_i^{In}})$
    for all planes $i$ do
        $\boldsymbol{\Delta_i^{In}} \leftarrow getNext\Delta(\boldsymbol{n_i}, \boldsymbol{\Delta_i^{In}})$
        $\boldsymbol{\Delta_i^{Out}} \leftarrow getNext\Delta(-\boldsymbol{n_i}, \boldsymbol{\Delta_i^{Out}})$
```

the obtained result from the original point, see Algorithm 6. Point clouds ($planePts$) that are corrected in this manner can be transformed into a mesh representation, which can be further transformed into a voxel representation.

4.3.2 Direct slicing of voxel-based models

The procedure for the direct slicing of voxel-based models is divided into two major steps: boundary extraction and generation of CNC instructions, see Figure 4.7.

The starting reconstructed volumetric model is a 3D array $\boldsymbol{\mathcal{V}}$, $\boldsymbol{\mathcal{V}} \in [0, 1]^{nVoxX \times nVoxY \times nVoxZ}$, where $nVoxX$, $nVoxY$ and $nVoxZ$ stand for the number of voxels in each of the directions. Already prior to the reconstruction, $nVoxZ$ was

Algorithm 6 `CorrectionByProjection`

```
Input:
    $planePts_i$ – set of all points assigned to every plane $i$
    $a_i, b_i, c_i, d_i$ – coefficients assigned to every plane $i$
Output:
    $planePts_i$ – set of all reallocated points assigned to every plane $i$

for all planes $i$ do
    $\boldsymbol{n_i} \leftarrow (a_i, b_i, c_i)$
    for all points $P$ in $planePts_i$ do
        $dist_P \leftarrow \frac{a_i x_P + b_i y_P + c_i z_P + d_i}{\sqrt{a_i^2 + b_i^2 + c_i^2}}$
        $P \leftarrow (P - dist_P \cdot \boldsymbol{n_i})$
```

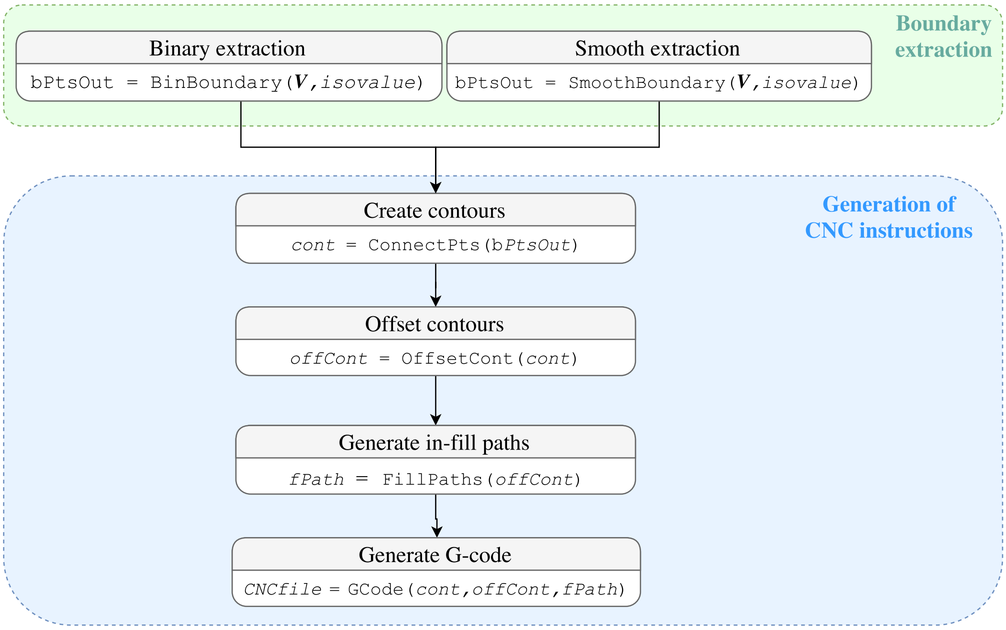


Figure 4.7: Main steps of the direct slicing: two proposed algorithms for the boundary extraction, followed by the multiple steps that lead to the correlated CNC instructions.

adjusted to the layer thickness of a particular AM process. With that, each of the xy-layers in $\boldsymbol{\mathcal{V}}$ can be considered as a layer of a model for the aimed AM process. Boundary voxels are extracted within each of these layers. Two alternative approaches are proposed for that: *Binary Extraction* and *Smooth Extraction*. Based on the extracted boundary points of each of the layers, closed contours are formed and slightly offset due to the width of the printing unit. The offset contours are then filled with paths, along which the interior of the contours are filled. The last step is a generation of the machine-interpretable CNC (computer numerical control) instructions. CNC files are often referred to as *G-codes* since G-code is one of the most widely used CNC programming language.

Binary Extraction

In the `BinaryExtraction` approach (Algorithm 7), all the voxels are categorised at the very beginning into two categories. If the attenuation value of the reconstructed voxel is above or equal to the *isovalue* (determined as described in the Section 4.2.3), the voxel gets a label 1, i.e. it is considered to be a part of the model. Otherwise, if the attenuation value is under the *isovalue*, the voxel is labelled with a 0, i.e. it is not considered to be a part of the model. After the binary categorisation, the actual boundary of each connected component (intersection line) is extracted with the Moore-Neighbouring algorithm, [Pradhan et al., 2010]. This algorithm starts from a random 1-labelled voxel with at least one 0-labelled neighbour and examines its neighbours in a predefined order. All encountered 0-labelled voxels are added to the layer boundary $bPts$. If, however, another 1-labelled voxel is encountered, its neighbours will be examined, starting from the last visited 0-voxel. This procedure is repeated until the starting voxel is not revisited. Final collected output boundary points on all layers are denoted by $bPtsOut$.

Algorithm 7 `BinaryExtraction`

```
Input:
    $\boldsymbol{\mathcal{V}}$ – 3D array of reals
    $isovalue$ – real number
Output:
    $bPtsOut$ – set of all boundary points

$bPtsOut \leftarrow \emptyset$
for all layers $l$ in $\boldsymbol{\mathcal{V}}$ do
    for all voxels $V$ in $l$ do
        if $\boldsymbol{\mathcal{V}}(V) \geq isovalue$ then
            $\boldsymbol{\mathcal{V}}(V) \leftarrow 1$
        else
            $\boldsymbol{\mathcal{V}}(V) \leftarrow 0$
    $bPts \leftarrow \emptyset$
    for each intersection line $L$ do
        $Vstart \leftarrow$ random $V$ related to $L$
        $bPts \leftarrow bPts \cup Moore(\boldsymbol{\mathcal{V}}, Vstart)$
    $bPtsOut \leftarrow bPtsOut \cup bPts$
```

Smooth Extraction

Due to the binary classification of voxels in the `BinaryExtraction`, the obtained border line highlights the underlying voxelised structure, i.e. it does not have a smooth structure (see Figure 5.13a). `SmoothExtraction` approach leverages this by adopting the *watersheding* segmentation method (presented in Section 2.4.3).

In the beginning, the centre points of all voxels V within one layer l are connected with their direct right, down and diagonal down-right neighbour (all three neighbour groups are denoted with $N(V)$), as depicted in Figure 4.8.

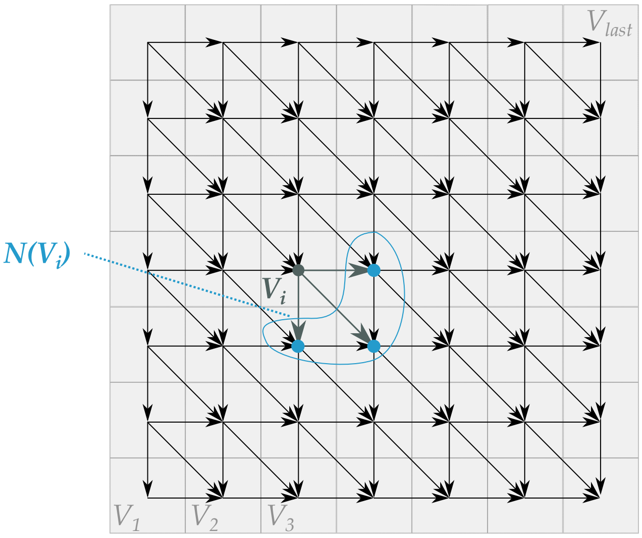


Figure 4.8: Imposing triangular connections between neighbouring voxels within one layer: each voxel centre is connected with their direct right, down and diagonal right-down neighbour.

The voxel attenuation values $\boldsymbol{\mathcal{V}}(V)$ are viewed as heights of a relief. A set of boundary points $bLoc$ is obtained by intersecting the relief with a plane which is parallel to the layer-plane at the *isovalue*-height. In addition to the boundary points, the starting and the ending points of each intersecting line between the relief and the plane are denoted. The final step is sorting of the boundary points, so that their traversal forms a boundary line. This is done by following the points of the intersection line L and connecting the segments. Each component of the sorted boundary points is denoted by $bPts$ and all collected output boundary points by $bPtsOut$. Algorithmic steps of the `SmoothExtraction` approach are presented in the Algorithm 8.

4.3.3 Transformation to mesh-based models

An alternative to the direct slicing of the voxel-based models is their transformation to the mesh-based models. This is realised by applying the Marching cubes algorithm with the user-defined *isovalue*, [Lorensen and Cline, 1987].

Algorithm 8 `SmoothExtraction`

```
Input:
    $\boldsymbol{\mathcal{V}}$ – 3D array of reals
    $isovalue$ – real number
Output:
    $bPtsOut$ – set of all boundary points

$bPtsOut \leftarrow \emptyset$
for all layers $l$ in $\boldsymbol{\mathcal{V}}$ do
    for all voxels $W$ in $N(V)$ do
        $bLoc \leftarrow \emptyset$
        for all voxels $V = (x_V, y_V, z_V)$ in $l$ do
            if $isovalue$ lies between $\boldsymbol{\mathcal{V}}(V)$ and $\boldsymbol{\mathcal{V}}(W)$ then
                $f \leftarrow \frac{isovalue - \boldsymbol{\mathcal{V}}(V)}{\boldsymbol{\mathcal{V}}(W) - \boldsymbol{\mathcal{V}}(V)}$
                $bLoc \leftarrow bLoc \cup \{V + f \cdot (W - V)\}$
                $L \leftarrow$ assemble intersection lines
    $bPts \leftarrow \emptyset$
    for each intersection lines $L$ do
        $bPts_L \leftarrow$ connected points of $bLoc$ according to $L$
        $bPts \leftarrow bPts \cup bPts_L$
    $bPtsOut \leftarrow bPtsOut \cup bPts$
```

Aside from transforming the initial voxel-based models, the inverted (reverse or negative) models are of significance for the further AM application fields too. AM increasingly impacts more conventional manufacturing methods, such as injection moulding. Moulds (casts) for non-printable materials such as plaster or metals are additively manufactured. For that, reverse volumetric models (with exchanged outside and inside) of minimal size are to be obtained. In order to do so, it is assumed that the volumetric model is composed out of only one material. Therefore, the voxels can be binary categorised into air and material.

Algorithm 9 describes the steps of the `ModelInversion` procedure. The first step is finding a bounding box of a volume by extracting the non-0 – valued voxels with the minimal and maximal coordinates. The attenuation values of the voxels within the computed bounds are then inverted by mirroring the corresponding values in $[0, 1]$ around the central 0.5 – value. In the final step, the inverted voxel-based model is transformed into a mesh by applying the Marching cubes algorithm with the inverted border value $1 - isovalue$. A set of vertices *ver* and faces *fac* describe the resulting mesh geometry.

4.3.4 Mesh simplification

The printing precision (resolution) differs not only among different AM processes but also for different printing axes within a single AM process. The precision in a printing plane (x- and y-axis) typically differs from the precision in the building direction (z-axis).

Algorithm 9 `ModelInversion`

```
Input:
    $\boldsymbol{\mathcal{V}}$ – 3D array of reals
    $isovalue$ – real number
Output:
    $ver, fac$ – sets of output vertices and faces

$BB \leftarrow$ bounding box of the initial model
for all voxels $V$ within $BB$ do
    $\boldsymbol{\mathcal{V}}(V) \leftarrow |\boldsymbol{\mathcal{V}}(V) - 1|$
$(ver, fac) \leftarrow MarchingCubes(\boldsymbol{\mathcal{V}}, 1 - isovalue)$
```

Therefore, the sought *mesh simplification* algorithm should eliminate the mesh vertices which are too close to each other according to the predefined precision limits in all three directions.

The proposed algorithm uses the *disjoint-set forests* (also known as union-find) data structure, [Cormen et al., 2001]. This data structure partitions a set of elements into non-overlapping subsets. Each subset is represented by a rooted tree, with the root being the representative element of the whole tree. Each element in a tree points only to its parent, with a root being its own parent. The main operations on the disjoint-set forests are *Make-Set*, *Find* and *Union*.

Make-Set operation initialises new trees with a single element.

Find operation returns the corresponding tree root by recursive parent-search.

Union operation combines two trees and returns a new common root.

Significant computational improvements are achieved by attaching the shorter tree (smaller rank) to the root of the taller tree (higher rank). The running time of all practical applications of the disjoint-set forests data structure is nearly linear in number of operations [Cormen et al., 2001], which makes the data structure very efficient.

In the `MeshSimplification` approach with the underlying disjoint-set forests data structure, the goal is to cluster the initial vertices into disjoint subsets. The vertices that belong to the same cluster (tree) are the ones which are too close in at least one of the printing axes. The root of a cluster is determined each time a new vertex is added to it. This is done by unifying the current root with the root of the newly added vertex and *merging* their coordinates. For the latter, the concept of weights is introduced. The weight of each vertex is at the beginning initialised to one. Each time two vertices unify, their weights are summed up and assigned to the new root. Merging of two vertices is then realised by their weighted linear combination. This merging concept disables large dislocations of roots of big clusters and thereby preserves the general shape.

The explicit steps of the `MeshSimplification` procedure are given in the Algorithm 10. At the very beginning, weights w are initialised. The main loop traverses through all initial faces fac and merges the associated vertices that are too close with respect to the limits $xLim$, $yLim$ and $zLim$. This process is repeated until all original vertices ver are not clustered, i.e. no merge is performed. The last step is extracting the vertices *Sver* and faces *Sfac* from the original *ver* and *fac* in order to obtain the simplified mesh. *Sver* are obtained by gathering the roots of all clusters; and *Sfac* by gathering the facets that have all three vertices from different clusters.

Algorithm 10 `MeshSimplification`

```
Input:
    ver, fac – sets of input vertices and faces
    $xLim, yLim, zLim$ – real numbers
Output:
    Sver, Sfac – sets of output vertices and faces

$\boldsymbol{w} \leftarrow$ 1-vector of size = number of vertices
repeat
    $merge \leftarrow false$
    for all faces $F$ in $fac$ do
        for each edge $(A, B)$ of face $F$ do
            if $find(A) \neq find(B)$ &
              $max\left\{\frac{|X_A - X_B|}{xLim}, \frac{|Y_A - Y_B|}{yLim}, \frac{|Z_A - Z_B|}{zLim}\right\} < 1$ do
                $Aroot \leftarrow find(A)$
                $Broot \leftarrow find(B)$
                $C \leftarrow union(Aroot, Broot)$
                $w_C \leftarrow w_{Aroot} + w_{Broot}$
                $ver(C) \leftarrow \frac{w_{Aroot} \cdot ver(Aroot) + w_{Broot} \cdot ver(Broot)}{w_C}$
                $merge \leftarrow true$
            end if
until $merge = false$
$Sver \leftarrow \emptyset$
for all vertices $V$ in $ver$ do
    if $V = find(V)$ then
        $Sver \leftarrow Sver \cup \{ver(V)\}$
$Sfac \leftarrow \emptyset$
for all faces $F = (A, B, C)$ in $fac$ do
    if find(A), find(B) and find(C) are pairwise disjoint then
        $Sfac \leftarrow Sfac \cup \{fac(F)\}$
```

4.4 Reduction of voids for droplet-based extrusion-AM

4.4.1 Detection and filling of general voids

The concept of detection and filling of general voids was published by Prša et al. in July 2015 at the International Conference on Computer, Information and Telecommunication Systems (CITS), [Prša et al., 2015b].

The tool for the general reduction of voids that are produced by arbitrary filling strategy is divided into four modules:

Parser reads a G-code file and extracts droplet coordinates, which resulted from the previously calculated filling strategy and other relevant parameters.

Analyzer sorts droplets in two steps. Firstly, droplets are layer-wise extracted. Secondly, droplets are classified into boundary droplets (contour) and interior filling droplets.

Hole Detection and Filling detects possible voids that the droplets may form, with the analysed droplets and the additional input parameters, and fills them afterwards.

G-code output writer produces a new G-code output file with the newly added droplets on each layer.

A flow chart of this process can be seen in the Figure 4.9. Two inputs of the program are the aforementioned G-code file and configuration parameters. One of these parameters is the overlapping threshold, which indicates how dense the additional droplets should be positioned, in terms of the maximally allowed overlapping surface with other droplets. The parser reads the initial droplet positions and the analyzer sorts the droplets. Afterwards, the detection of voids takes place and, together with the configuration parameters input, voids are filled with additional droplets. Finally, the initial G-code file is enriched with additional droplets.

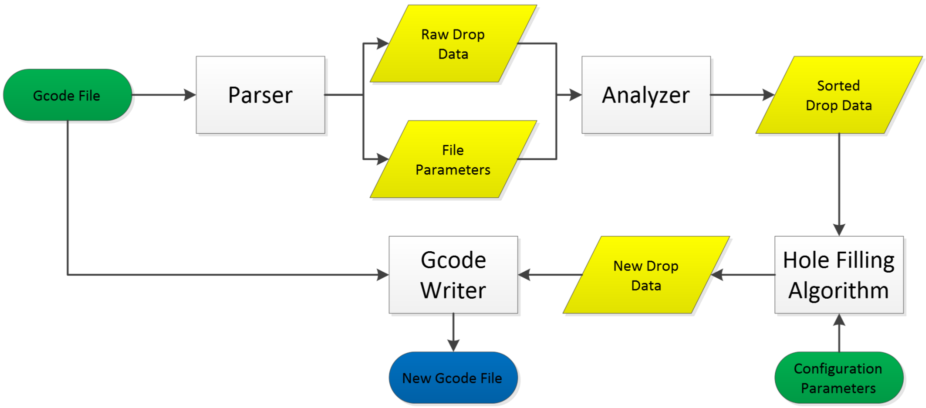


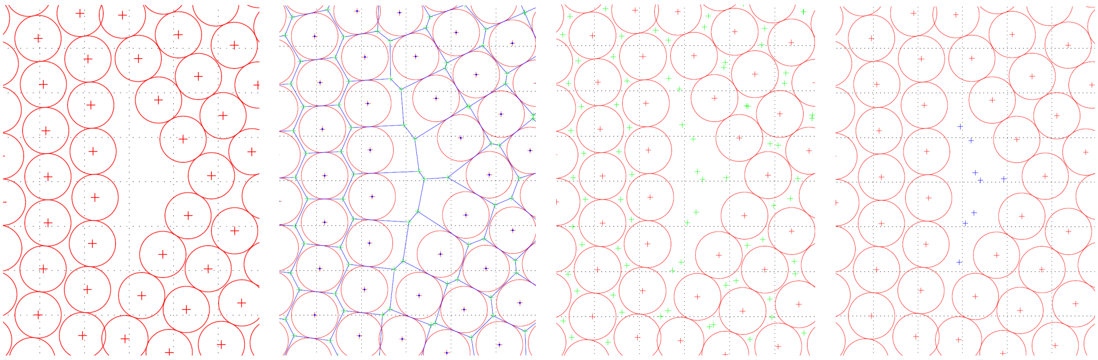
Figure 4.9: Process of detection and filling of general voids.

While standard infill strategies are being uniformly applied without considering the distinct areas, this approach tends to individualise the process of filling, by repairing the under-fills only where necessary. Moreover, with controlling the threshold value, the user can control void filling and possible induced overlaps with only one parameter and with that generate filling that is optimal for a various AM systems.

Once sliced, 3D spheres can be represented in a 2D level as equal-sized circles. With a constant radius of the circles, droplets within a slice can be described as points with two coordinates, and therefore one slice can be treated as a cloud of points. In order to detect a hole, Voronoi Diagrams have been applied. A Voronoi diagram is a result of the plane partitioning into n convex polygons out of n generating points, with the property that every point within a polygon is closer to its own polygon's generating point than to any other generating points. Once Voronoi polygons are obtained, their vertices can be extracted. If the distance from one generating point to any of its vertices is above some threshold value, a hole is detected.

Detection Process

After the droplet data has been analysed and sorted according to the layers and boundary/infill classification, the detection process can start. Voronoi diagram is launched for a cloud of points for each of the layers that do not belong to the layer boundary. Upon extraction of the vertices of each Voronoi polygon, the distances from the vertices to the nearest generating point are calculated. If the distance surpasses a certain threshold value, the corresponding vertex is marked and will be processed in the void filling part. This process is depicted in the Figure 4.10.

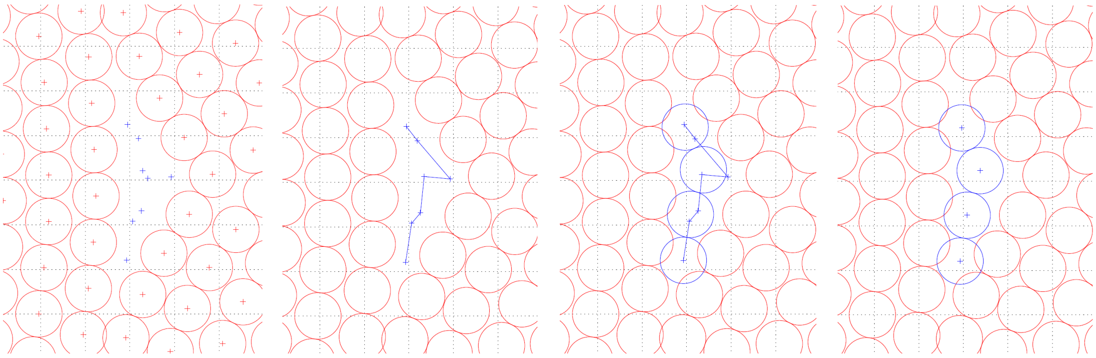


(a) Initial points. **(b)** Division into Voronoi polygons. **(c)** Extracted Voronoi vertices. **(d)** Marked far off vertices.

Figure 4.10: Simulation of void detection via Voronoi diagrams.

Filling Process

Voids that are identified as far off Voronoi vertices serve as a starting point for the filling process. Vertices that form one void are sorted as follows: the centre of the void has been calculated, and the vertex that is the furthest from it becomes a starting point. The closest vertex (among remaining vertices) to the starting point, has been searched and added behind the starting point. This procedure is repeated until there are no more vertices left. In case the distance between two vertices is smaller than a threshold value, one of them will be eliminated. Sorted vertices are connected with a line and additional droplets are set along this line. The positioning of new droplets is influenced by the other droplets of the layer in a way that the new droplets have minimal overlap with the original droplets. Due to this, the positioning of the additional droplets is allowed to be slightly aside of the previously calculated line. The hole filling process is depicted in the Figure 4.11.



(a) Marked far off vertices. **(b)** Line that connects the vertices. **(c)** Positioning of the droplets along the line. **(d)** Final result.

Figure 4.11: Simulation of the void filling process.

4.4.2 Definition and filling of pointed ends

A method for definition and filling of pointed ends was published at the ASME International Mechanical Engineering Congress & Exposition (IMECE) by Prša et al. in November 2014, [Prša et al., 2014a] and at the IEEE International Conference on Robotics and Biomimetics (ROBIO) by Prša et al. in December 2014 [Prša et al., 2014b].

The method for detecting and solving the problem of under-filled pointed ends starts with the placing of droplets along the contours, so that the first droplets are positioned at the end points of a contour segment. They are iteratively followed by their neighbours in the direction of the segment centre. The novelty of the suggested method are additional droplet inserts and removals. Additional droplets are to be placed in the angle bisector as long as the overall over-fills do not surpass the newly filled void area by the inserted droplet. After the insertion of each droplet, the overall error (faults induced either by voids or over-fills) is to be obtained. Further insertion continues as long as the overall error decreases. In the end, the initial droplets in the very tip of a pointed end that overlap too much are removed.

Sharp angles α are defined as the angles that are smaller than 60°. The critical area, on which the corresponding under-fills and over-fills are measured, is defined as presented in Figure 4.12. The surface P of the area is calculated with the following equation:

$$P = \frac{16}{\tan\left(\frac{\alpha}{2}\right)} \tag{4.2}$$

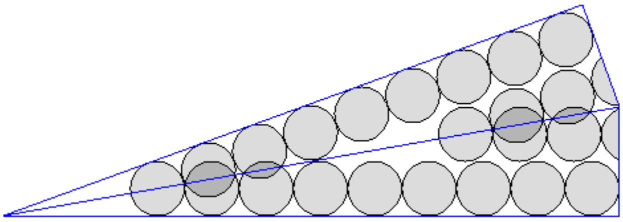


Figure 4.12: Critical area: a quadrilateral enclosed by two outer contours of a sharp corner and their two orthogonal segments, each of them the size of two droplet diameters long.

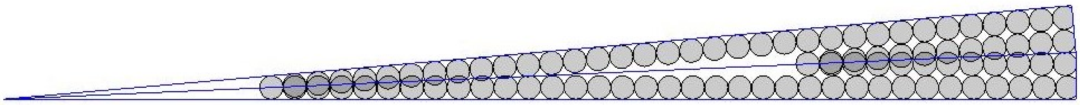
The formulae for calculating the errors are given in the Equations (4.3). While P stands for the total surface of the critical area, P_- and P_+ are the surfaces of the under-filled and over-filled areas within the critical area. The over-filled area P_+ is in turn divided into P_+^{init} and P_+^{add}. The first one is the area of the initial over-fills originating from the overlapped droplets that are placed along the contours. The latter one represents the area of the overlaps induced by the additional droplets that are placed along the angle bisector. Depending on those values and on the total surface area of the voids around the each newly inserted additional droplet, under-fill Error (E_-), over-fill Error (E_+) and overall Error (E) can be calculated and embedded into the algorithm.

$$\begin{aligned}
E_- &= \frac{P_-}{P}, & E_- &\in [0,1] \\
E_+ &= \frac{P_+}{P} = \frac{P_+^{init} + P_+^{add}}{P} \\
&= \frac{P_+^{init} + \sum_{k=1}^{opt} \omega_k P_{+,k}^{add}}{P}, & E_+ &\in [0,1] \\
\text{with } \omega_k &= \begin{cases} 0 & Void_k > P_{+,k}^{add} \\ 1 & Void_k \leq P_{+,k}^{add} \end{cases} \\
E &= E_- + E_+
\end{aligned} \tag{4.3}$$

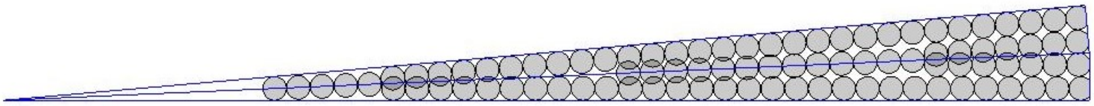
The under-fill and over-fill errors have the same weight in the overall error calculation since their negative impacts on the overall surface quality are alike. A certain amount of overfill leads to material surpluses on the part sides, while the same amount of voids leads to the same amount of leaks, both of which results in the uneven final surface and undesired geometry-shapes. The double-filled area P_+ is separated into two parts.

The first part consists of the initial over-fills prior to execution of the algorithm (P_{+}^{init}). The second part is formed by the additional over-fills that are generated with each droplet insertion (P_{+}^{add}). Single additional over-fills $P_{+,k}^{add}$ are weighted depending on the surrounding voids ($Void_k$), measured within a circle with a radius of a 3/2 of the droplet radius. If a single additional over-fill is smaller than the surface of the surrounding voids, it will be consumed by the surrounding voids.

The starting infill error E is calculated before any insertion. By inserting additional spheres in the sharp angle bisector (while not surpassing the outer boundaries), the infill error is being reduced (since E_{-} decreases), until newly inserted spheres do not start interfering with the rest of the spheres. The stopping criterion is the first inserted sphere, for which the error E starts growing (E_{+} increases drastically). In this way, the voids in the angular bisector area are reduced, but the over-fills are increased as well. As both under-fills and over-fills cannot be avoided (due to the ballistic shape of the droplets) and are equally undesirable, there is a certain trade-off between the under-fills and over-fills, such that the overall infill deviation is minimised. The final step of the algorithm is the deletion of superfluous spheres. The spheres that are placed along the contour segments and overlap for more than 50% of their surface are removed. The results of this procedure for the 5° pointed end, with eight insertions and four deletions, are shown in Figure 4.13.



(a) Before applying the insertions and deletions.



(b) After applying the insertions and deletions.

Figure 4.13: A critical area of a 5° pointed end before and after applying the proposed procedure.

5 Realisation

This chapter presents the realisation of the concepts that were introduced in the previous Chapter 4. The X-ray based digitization unit will be presented in Section 5.1 and the post-processing unit in Section 5.2. A tool for reduction of voids follows in Section 5.3.

All presented algorithms were tested on a 64 Bit PC with Intel i7-4700HQ CPU, 8GB RAM and NVIDIA GeForce GTX 880M graphics card using Windows.

5.1 X-ray based digitization

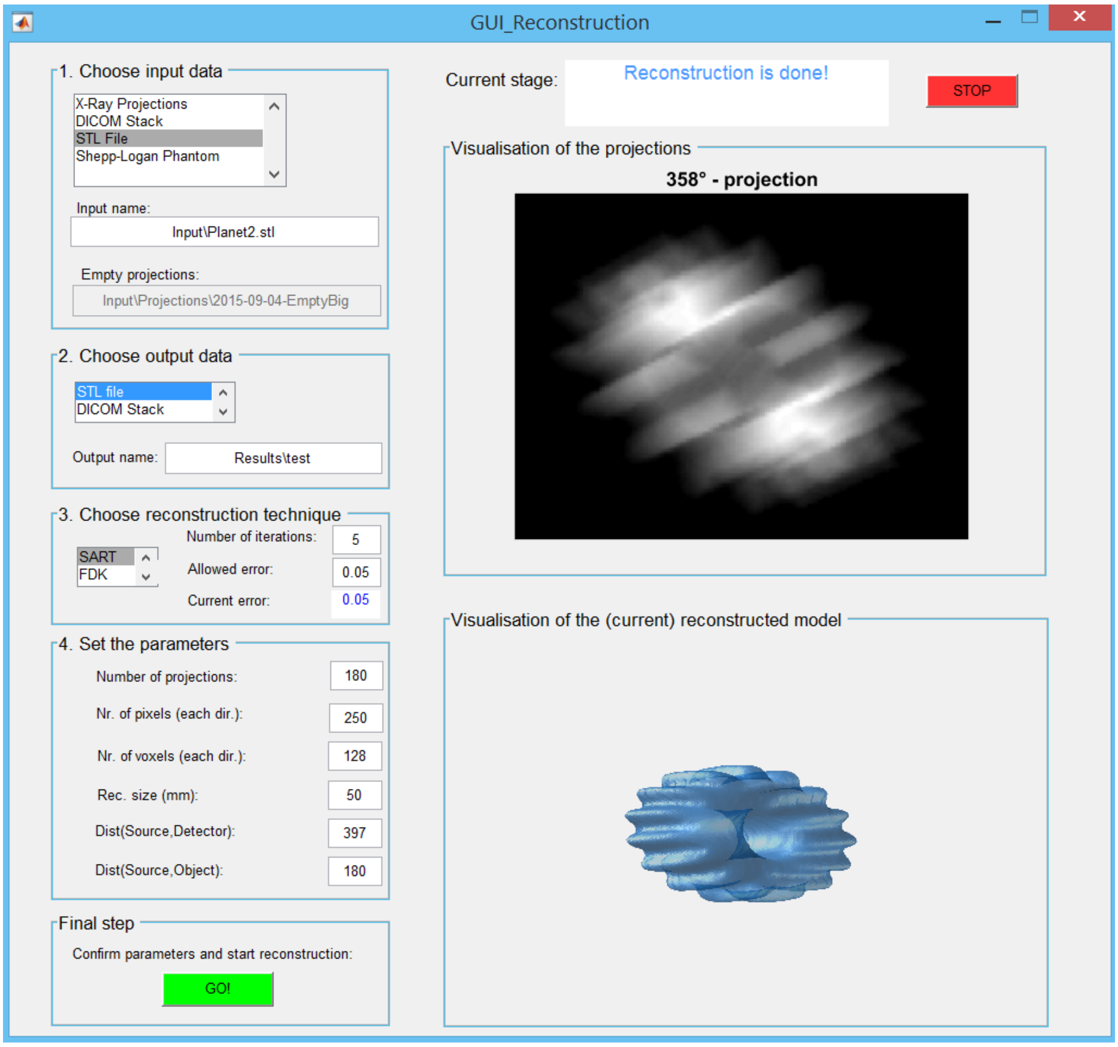
5.1.1 Reconstruction unit

The realisation of the reconstruction unit relies on the concepts introduced in Sections 4.1 and 4.2. Standard mode enables the reconstruction of sets of previously collected X-ray projections. X-ray simulator mode enables the simulation of the virtual projection data for DICOM stacks, STL files and Shepp-Logan phantom models. Implemented reconstruction algorithms are SART and FDK, with the computations running in parallel on GPU.

These algorithms have been implemented in Matlab using its toolboxes: Image Processing, Parallel Computing; and Statistics and Machine Learning. Additional external Solid Geometry toolbox (SG-Lib/VLFL-Lib 3.7, [Lueth, 2017]) has been used as well. In order to gain from the GPU-acceleration (Parallel Computing toolbox), a CUDA - enabled NVIDIA graphics card with a compute capability 2.0 or higher is required. In case of a positive answer from an automated check for Matlab-supported GPU, computations will be automatically accelerated. Without the compatible GPU, all the routines will still function properly on the CPU.

In addition to executing different reconstruction modes by function calls, a graphical user interface (GUI) is provided as well and will be presented here in more detail. The preparation for the reconstruction is divided into four sections: input data, output data, reconstruction technique and parameters set-up, see Figure 5.1.

In the *input data* section, users can define the name of the input data and choose between X-ray projections, DICOM stacks, STL files and Shepp-Logan phantoms. In case of reconstructing a set of X-ray projections, a set of empty projections has to be provided as well. In the *output data* section, users can define the name of the output data and choose between STL files and DICOM stacks. Users can choose whether the reconstruction is carried out via SART or FDK in the *Reconstruction technique* section. In the SART case, the number of iterations has to be set as well. If the virtual projections are to be reconstructed with SART, users can set the allowed error. In this case, the intermediate (current) errors will be compared against the allowed error. As



Input name: Input\Planet2.stl

Empty projections: Input\Projections\2015-09-04-EmptyBig

Output name: Results\test

Number of iterations: 5

Allowed error: 0.05

Current error: 0.05

Number of projections: 180

Nr. of pixels (each dir.): 250

Nr. of voxels (each dir.): 128

Rec. size (mm): 50

Dist(Source,Detector): 397

Dist(Source,Object): 180

Confirm parameters and start reconstruction:

Current stage: Reconstruction is done!

Figure 5.1: Matlab GUI realisation of the reconstruction unit.

the intermediate errors sink and reach the allowed error, the reconstruction procedure will be automatically stopped. In the *parameter set-up* section, the following parameters can be set: number of projections (rotational angles), number of pixels (detector size), number of voxels and their sizes, distance between the X-ray source and the detector (dis_{SD}) and distance between the X-ray source and the object (dis_{SO}).

After setting up all the parameters, the reconstruction process can start. Two visualisation units provide insights into the reconstruction process. The first one visualises the projections, while the second one provides the visualisation of a current reconstructed model. If the quality of the model is satisfactory, users can preliminary stop the reconstruction procedure and immediately obtain the output model. Otherwise, the reconstruction will continue as long as the initial parameter configuration requires.

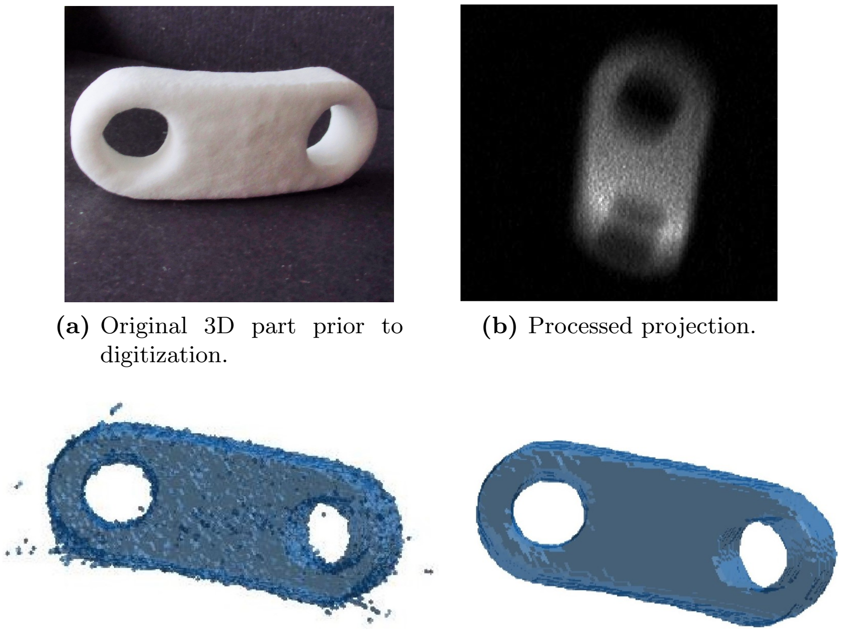
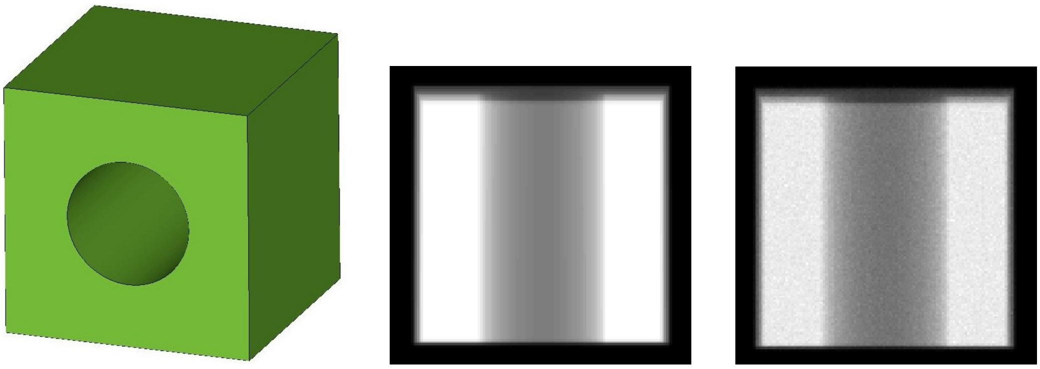

(a) Original 3D part prior to digitization.
(b) Processed projection.
(c) 3D reconstructed model.
(d) Post-processed model with median filter.

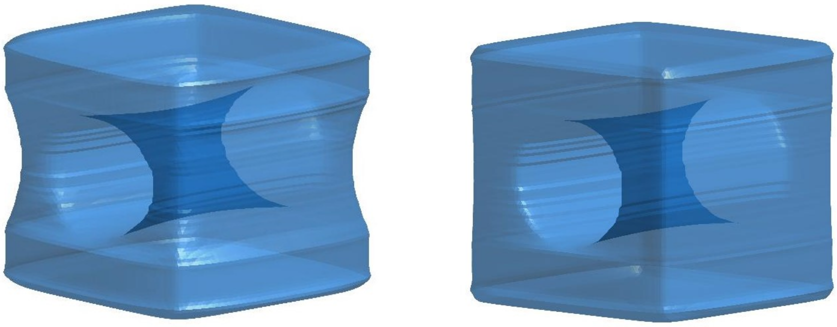
Figure 5.2: 3D reconstruction of a kinematic link from real X-ray projections with FDK algorithm.

In order to demonstrate the effectiveness of the reconstruction unit, a plastic (PA) kinematic link with a longitudinal length of 45 mm (depicted in Figure 5.2a) has been digitized. 180 X-ray projections and corresponding empty projections have been captured by Leybold X-ray apparatus (35 kV and 1 mA). dis_{SD} was set on 397 mm and dis_{SO} on 227 mm. The projections had dimensions of 300×300 pixels with 16 bit depth. An example of one projection is depicted in Figure 5.2b. The number of voxels in each dimension was 128 with a size of 2.56 mm. The result of the 3D reconstruction via FDK algorithm with this parameter setting can be seen in Figure 5.2c. After applying a median filter of the size 3×3, salt-and-pepper noise is eliminated. The final solution is depicted in Figure 5.2d. Processing of the projections, FDK reconstruction and post-processing lasted approximately 15 s on the machine introduced at the beginning of the chapter.

For the demonstration of the X-ray simulator, a 3D STL model of a cube with a cylindrical bore ($20 \times 20 \times 20$ mm, $r = 5$ mm) has been used (depicted in Figure 5.3a). 180 virtual projections with $dis_{SD} = 397$ mm and $dis_{SO} = 187$ mm have been made. The projections had dimensions of 250×250 pixels with 16 bit depth. An example of one projection is depicted in Figure 5.3b. The same projection with additional noise is depicted in Figure 5.3c. The number of voxels in each dimension was 128 with the size of 0.26 mm. The intermediate result of the 3D reconstruction after two iterations of SART algorithm with this parameter setting (without noise) can be seen in Figure 5.3d. The final reconstructed model after five iterations is presented in Figure 5.3e.



(a) Original STL model.　**(b)** Virtual projection.　**(c)** Virtual projection with added noise.



(d) Intermdiate 3D reconstructed model after two iterations.　**(e)** Final 3D reconstructed model after five iterations.

Figure 5.3: 3D reconstruction of a cube with a cylindrical bore from virtual X-ray projections with SART algorithm.

Voxelisation of the 3D model, simulation of the virtual projections, their processing and SART reconstruction lasted approximately 4.5 min on the machine introduced at the beginning of the chapter.

The reconstruction unit was successfully tested for a variety of other parts and virtual models (for all modes and reconstruction algorithms) on 64 Bit PC with NVIDIA GeForce GTX 880M graphics card using Windows with Matlab 2016b, 64 Bit PC with AMD Radeon HD 7570 graphics card using Windows with Matlab 2016b, and OSX 10.11 El Capitan with NVIDIA GeForce 9400M graphics card with Matlab 2017a.

5.1.2 X-ray parameters database

The automated setting of X-ray tube parameters is realised via a look-up table, as described in Section 4.2.3. The look-up table connects commonly used materials and several standard object sizes in RP with suitable X-ray parameters.

A look-up table is realised for cylindrically shaped objects of three different sizes: small cylinder ($r = 20\,\text{mm}$), medium cylinder ($r = 40\,\text{mm}$) and large cylinder ($r = 60\,\text{mm}$). The height of all three cylinders is 20 mm. The cylinders have been printed with four materials: PA, photopolymer, wood and ABS (see the material specification in Table 5.1). All 12 printed cylinders can be seen in Figure 5.4.

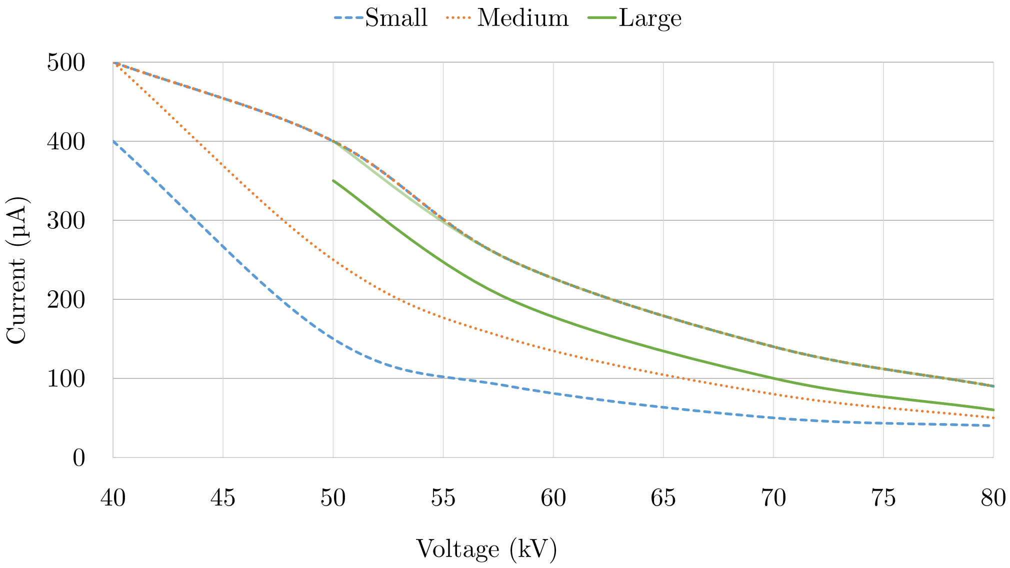
Table 5.1: Materials description.

Nr.	Identifier	Full description	Density (kg/m^3)
1	PA	polyamide 12 (PA 2200)	930
2	photopolymer	high temperature $RGD525$	1178
3	wood	wood composition	773
4	ABS	acrylonitrile butadiene styrene	1052

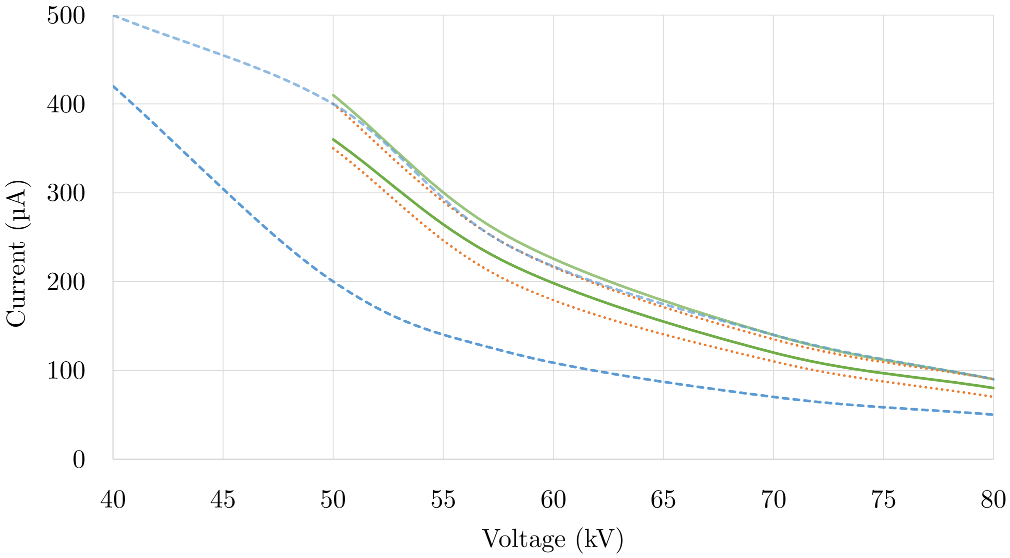
Figure 5.4: Three diferent-sized cylinders out of four different materials from left to right: photopolymer, ABS, wood and PA.

Test scans of all 12 cylinders were made with Wenzel Desktop-CT exaCT XS (max. 80 kV and 500 µA) in two steps. In the first step, lower and upper limits of I were iteratively searched for each of the discrete V-steps, so that the reconstruction result with the underlying projections is satisfying (according to the $20 - 80\%$ rule introduced in Section 4.2.3). V-steps were: $40, 50, 60, 70$ and 80 kV. In the second step, lower and upper limits of V are iteratively searched for each of the discrete I-steps, which were: $100, 200, 300, 400$ and 500 µA. The obtained limits are depicted in the Figures 5.5a (PA), 5.5b (photopolymer), 5.6a (wood) and 5.6b (ABS). The results show that with a higher radius of cylinders a range of acceptable parameters is narrowing. Moreover, since the density of the standard RP materials is relatively similar, the graphs are similar as well.

Finally, a database with X-ray parameters is presented in the Table 5.2. Single parameter sets are obtained by taking the average V with the corresponding average I values for each material and cylinder size. Due to the X-ray tube limitations of desktop X-ray apparatuses, several alternative sets of parameters are provided as well.

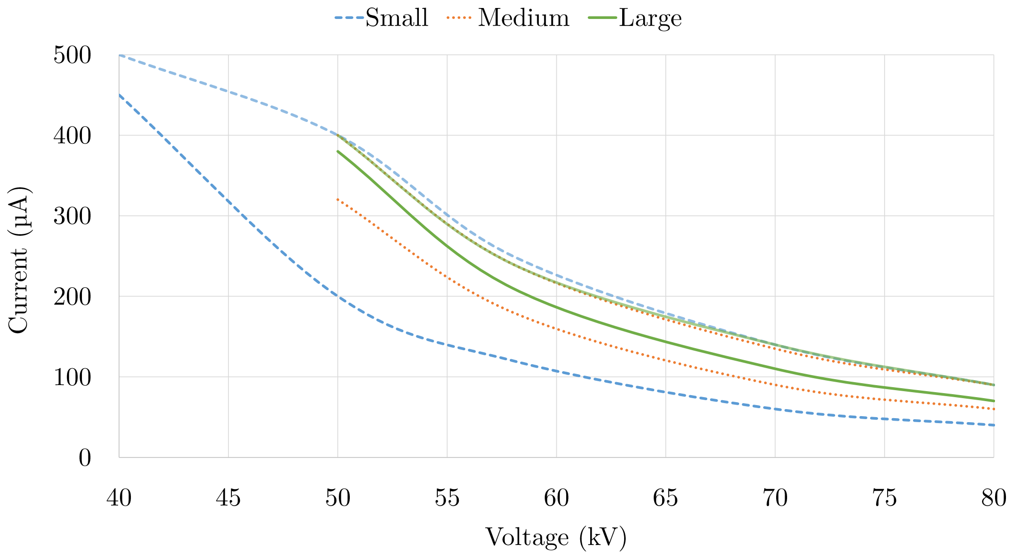


(a) Material: **PA**. All upper limits are almost overlapping.

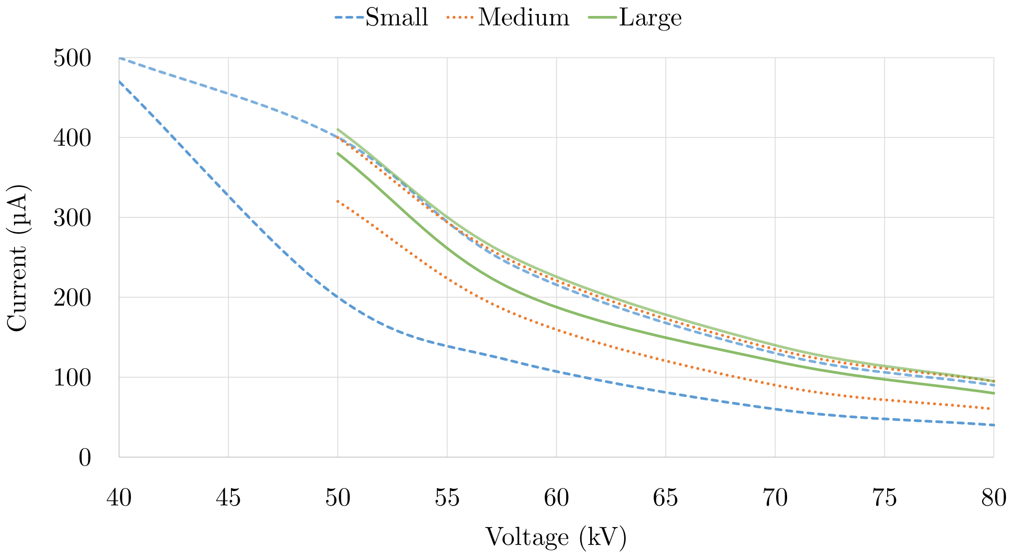


(b) Material: **photopolymer**.

Figure 5.5: The obtained limits for small (blue dashed lines), medium (red dotted lines) and large (full green lines) cylinders. The areas of the acceptable V and I parameters are between the corresponding matching lines.



(a) Material: **wood**. Upper limits for medium and large cylinders are almost overlapping.



(b) Material: **ABS**.

Figure 5.6: The obtained limits for small (blue dashed lines), medium (red dotted lines) and large (full green lines) cylinders. The areas of the acceptable V and I parameters are between the corresponding matching lines.

Table 5.2: Database which connects commonly used materials and object sizes with suitable X-ray tube parameters in (kV, μA).

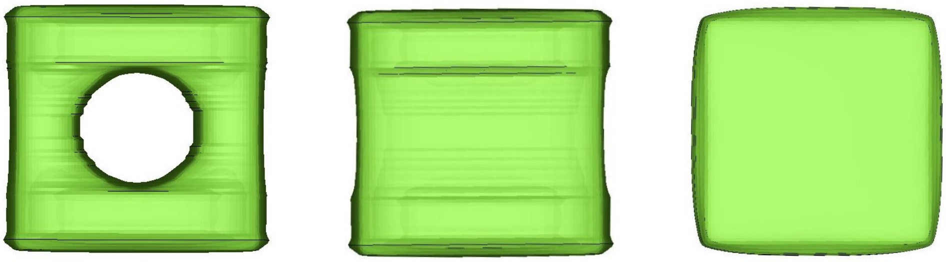
Object specifications	Single entry	Alternative entries
PA, $r = 20\,\text{mm}$	$(60, 170)$	$(45, 380)$, $(75, 100)$
PA, $r = 40\,\text{mm}$	$(60, 200)$	$(50, 280)$, $(75, 130)$
PA, $r = 60\,\text{mm}$	$(65, 175)$	$(52, 380)$, $(75, 100)$
photopolymer, $r = 20\,\text{mm}$	$(60, 180)$	$(45, 350)$, $(75, 110)$
photopolymer, $r = 40\,\text{mm}$	$(65, 170)$	$(52, 380)$, $(70, 120)$
photopolymer, $r = 60\,\text{mm}$	$(65, 185)$	$(50, 400)$, $(75, 100)$
wood, $r = 20\,\text{mm}$	$(60, 180)$	$(40, 480)$, $(70, 80)$
wood, $r = 40\,\text{mm}$	$(65, 150)$	$(50, 320)$, $(70, 90)$
wood, $r = 60\,\text{mm}$	$(65, 175)$	$(50, 400)$, $(80, 80)$
ABS, $r = 20\,\text{mm}$	$(60, 180)$	$(45, 400)$, $(70, 70)$
ABS, $r = 40\,\text{mm}$	$(65, 160)$	$(50, 350)$, $(80, 90)$
ABS, $r = 60\,\text{mm}$	$(65, 180)$	$(55, 300)$, $(80, 95)$

5.2 Post-processing unit

The realisation of the post-processing unit was carried out with Matlab 2016b.

5.2.1 Recognition and correction of planes

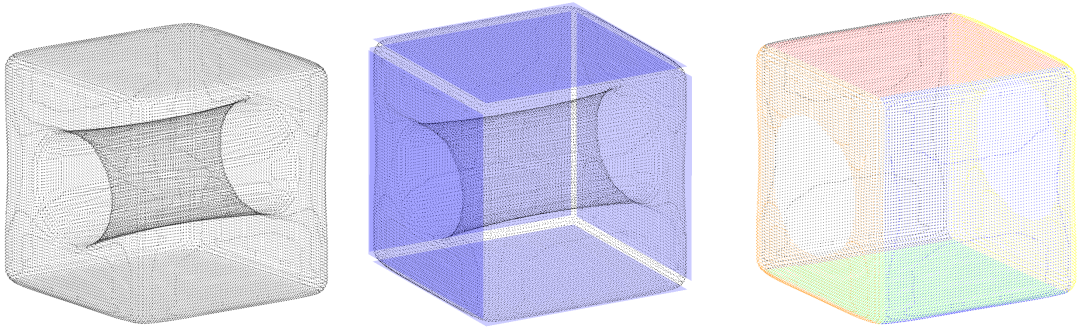
The methods for the recognition and correction of planes (introduced in Section 4.3.1) were implemented with the help of the Matlab's Computer Vision System Toolbox. In order to test and compare all the methods, a model of a cube with a cylindrical bore was reconstructed (SART with 5 iterations, 180 projections, 128^3 voxels of 0.26^3 mm size). The reconstructed model before any plane correction can be seen in Figure 5.7. One can notice slightly rounded edges in the bottom and top face, as well as shrank side faces along the bore.



(a) Front view. (b) Side view. (c) Top view.

Figure 5.7: Reconstructed model prior to plane correction.

The first step was plane detection. For that, the vertices from the mesh model have been extracted and a PC model was obtained. Afterwards, mathematical models of planes in the PC model were searched. The plane detection was realised via *pcfitplane* function from the Computer Vision System Toolbox. The two main inputs of the function are: the PC model and *MaxDist* from the plane to any inlier point. *MaxDist* was set on four times the voxel length and it was used as a limit distance within all stages of detection and correction. The *isovalue* was set on 0.35. The last step of the detection process was model splitting into planes by assigning points to detected plane models. The plane detection process is shown in Figure 5.8.



(a) PC model.

(b) PC model (black) with detected planes (violet). The front plane is hidden for better visibility.

(c) PC model split into planes: points that are assigned to one plane are of one colour.

Figure 5.8: Plane detection process.

Regarding the correction of the planes, three different algorithms have been compared against: Direct Voxel Correction, Correction by Projection and Smearing Correction (presented in Section 4.3.1).

In the *Direct Voxel Correction*, the attenuation value of the voxels is for each plane decreased (increased) if they are positioned inside (outside) the model (as described in Section 4.3.1). For the correction step that implements the functions *reduce* and *increase* in the Algorithm 4, two different approaches have been tested: binary and smooth (see Figure 5.9). In the first one, the correction was carried out binary – on either 0 or 1. In the latter one, the voxels V are corrected smoothly depending on their distance d_s to the plane:

$$\begin{aligned} d_s &= \frac{MaxDist - dist}{MaxDist} \\ \boldsymbol{\mathcal{V}}^{new}(V) &= \begin{cases} 0 & d_s < 0 \\ 3d_s^2 - 2d_s^3 & 0 \le d_s \le 1 \\ 1 & d_s > 1 \end{cases} \end{aligned} \tag{5.1}$$

The results of the Direct Voxel Correction are depicted in Figures 5.10a and 5.10b. The binary approach indicates unsmooth jumps on the surface texture. These jumps are

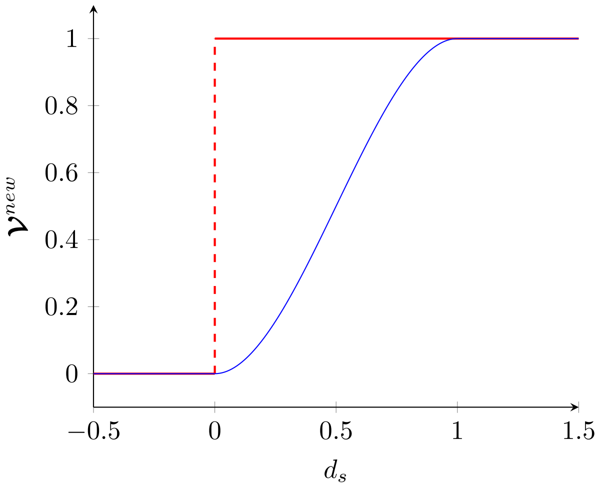
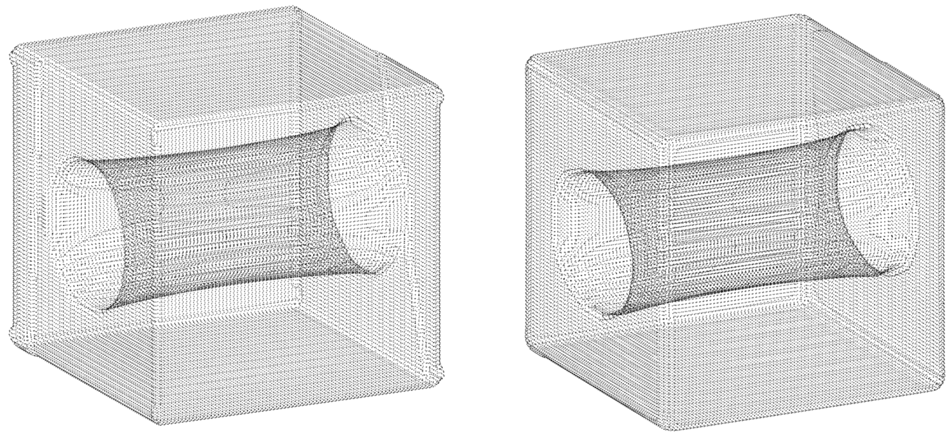


Figure 5.9: Comparison of two voxel correction approaches: binary step (in red) and smooth step (in blue).

evened with the smooth approach. However, the bordering areas between corrected and uncorrected parts (around the bore), as well as between two different corrected planes (edges), are not well connected. These issues are best visible in the corresponding mesh model, see Figures 5.11b and 5.11c. The reason for that might lie in the voxel correction approach, which is independent of the voxel's neighbourhood.

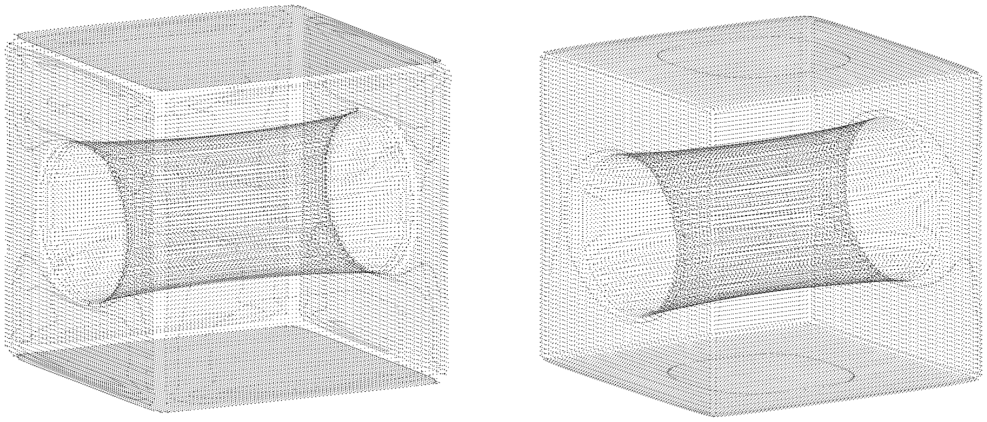
In the *Correction by Projection*, the extracted plane points were orthogonally projected on the corresponding planes. The results are shown in Figure 5.10c. With that, it is assured that the points are lying exactly on the planes. However, the connection of the bordering areas is still not solved. Besides, the transformation from the corrected PC models to the printable mesh models is troublesome. Different approaches have been considered, such as Delaunay triangulation on convex hulls, boundary search on non-convex alpha shapes, Power Crust algorithm [Amenta et al., 2001] and Poisson surface recognition [Kazhdan et al., 2006]. The Delaunay triangulation is not suitable due to its restriction to convex hulls and Power Crust algorithm is not suitable due to its long computational time. Poisson surface recognition algorithm was not tried out since it has a complex structure and its implementation would surpass the scope of this thesis. The best results were achieved with an alpha shape boundary. However, non-convex parts were still causing difficulties (see Figure 5.11d inside the bore). The additional voxelisation and smoothing of the voxel model partly evened the non-convex parts, but it also rounded the sharp edges, as presented in Figure 5.11e.

In the *Smearing Correction*, neighbouring voxels along the plane normals were calculated and iteratively corrected by smearing. In the correction step, neighbouring voxels were smoothly corrected with a linear combination of their current attenuation value and the attenuation value of the current voxel. Two different approaches have been tested: performing all corrections at once at the end of one iteration and performing the corrections immediately (in the traversal order). No significant difference was noticed. The



(a) Direct Voxel Correction with binary step.

(b) Direct Voxel Correction with smooth step.



(c) Correction by Projection.

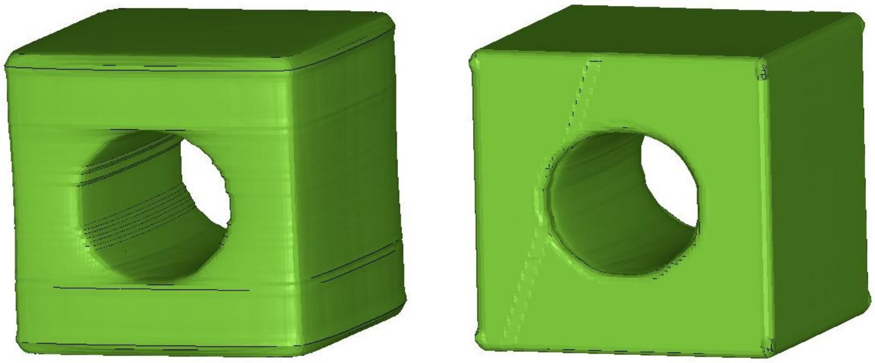
(d) Smearing Correction.

Figure 5.10: Corrected PC models.

final results of the Smearing Correction after 10 iterations are shown in Figure 5.10d. The resulting mesh model is depicted in Figure 5.11f.

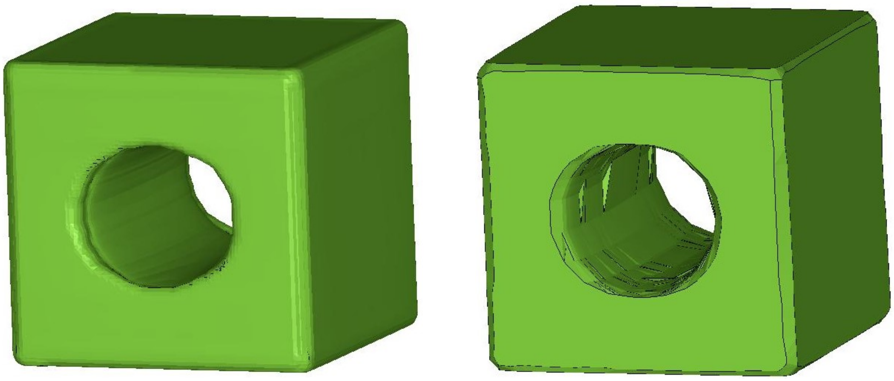
Concavity issues (after the Direct Voxel Correction and Correction by Projection) and the issues with the connection of the bordering areas (after the Correction by Projection) are with the Smearing Correction resolved and the final model is favourable. The detection and correction with all three algorithms for all six planes on models with approximately 50000 points lasted for <45 s on the machine introduced at the beginning of the chapter.

The Smearing Correction was employed on another reference object (presented in 4.4). After scanning and the FDK reconstruction with Wenzel Desktop-CT exaCT, the model contained ring artefacts. They were successfully corrected with the proposed method, see Figure 5.12.



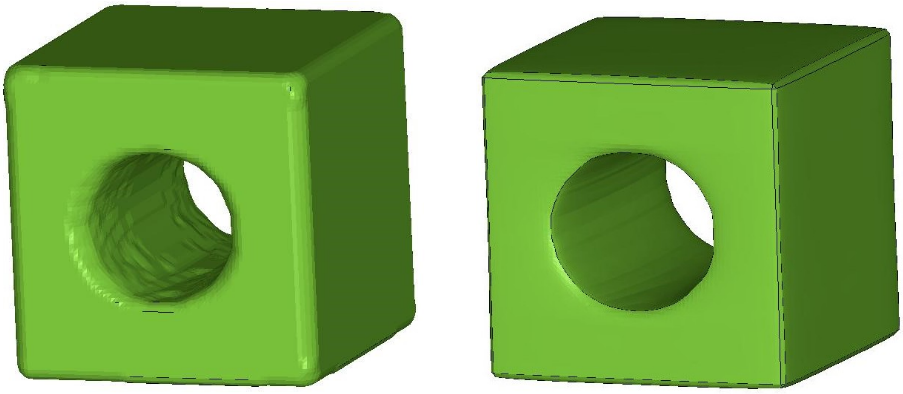
(a) Model before correction.

(b) Direct Voxel Correction (binary step).



(c) Direct Voxel Correction (smooth step).

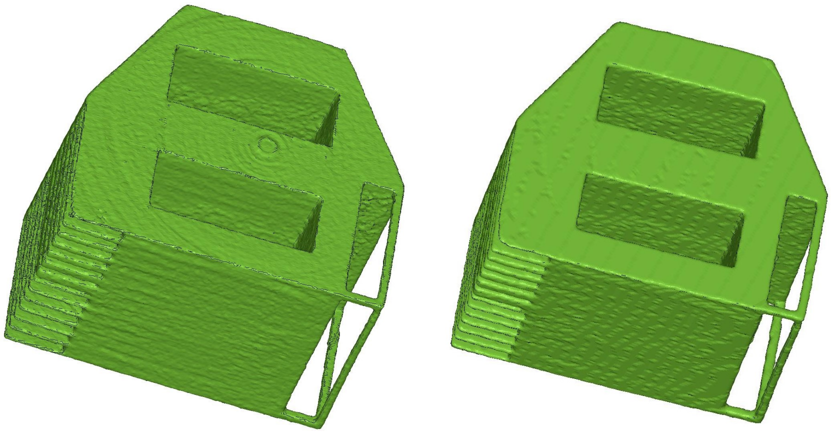
(d) Correction by Projection mesh obtained after direct transformation from PC.



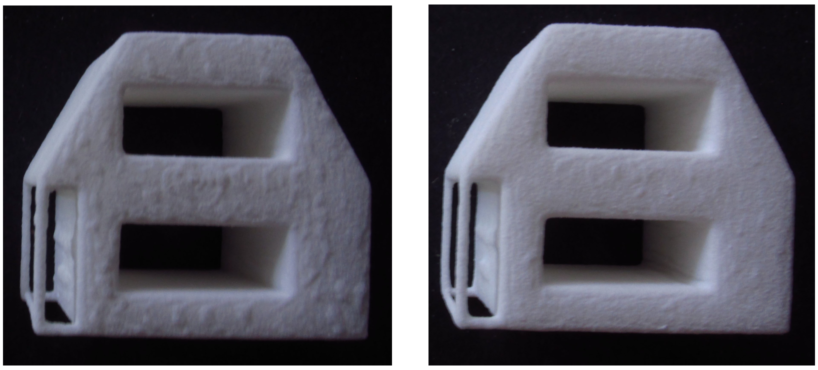
(e) Correction by Projection mesh obtained after intermediate voxelisation.

(f) Smearing Correction.

Figure 5.11: Final corrected mesh models.



(a) Reconstructed model prior to correction. **(b)** Reconstructed model after correction.



(c) 3D printed object without applied correction. **(d)** 3D printed object with applied correction.

Figure 5.12: Correction of ring artefacts by Smearing Correction. Upper two images show the model comparison and downer two the comparison between the corresponding 3D printed objects (printed with *EOS Formiga P 100*).

Restrictions

The final Smearing Correction approach has several restrictions:

- There exists no guarantee on the model's size preservation.
- The algorithm provides reasonable results only for the geometries which have planes orthogonal to the inner geometries.

5.2.2 Direct slicing

The methods for direct slicing were implemented with the help of Matlab's Image Processing Toolbox and Mapping Toolbox. As introduced in Section 4.3.2, direct slicing is carried out within each slice in two steps: boundary extraction and generation of machine-interpretable instructions. Regarding the boundary extraction, two different algorithms have been introduced: *Binary Extraction* and *Smooth Extraction*.

The final corrected mesh (with Smearing Correction) of the cube with a cylindrical bore was taken as a test model for comparison, see Figure 5.11f. For direct slicing in the direction along the bore, layer thickness was set on 0.27 mm and *isovalue* on 0.35. A comparison between two boundary extraction approaches can be seen in Figure 5.13. The obtained borderlines with Binary Extraction overemphasise the underlying voxelised structure. Improved smoother border lines are achieved by applying the Smooth Extraction.

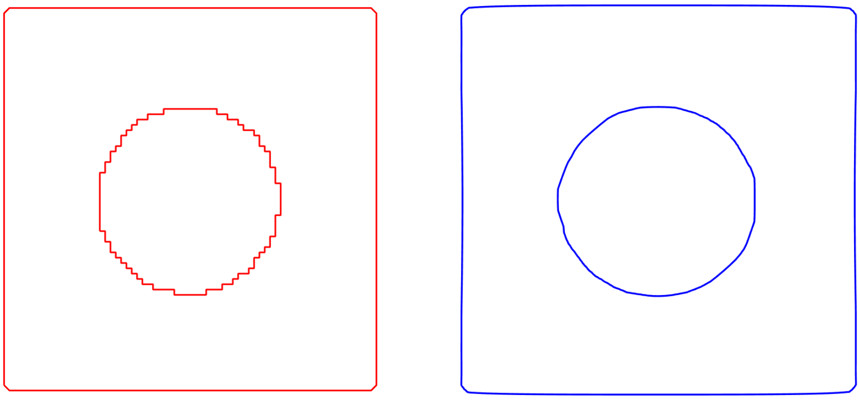

(a) Binary extraction.
(b) Smooth extraction.

Figure 5.13: Comparison of two boundary extraction approaches within one slice.

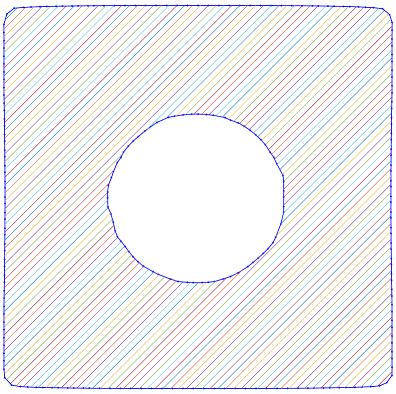


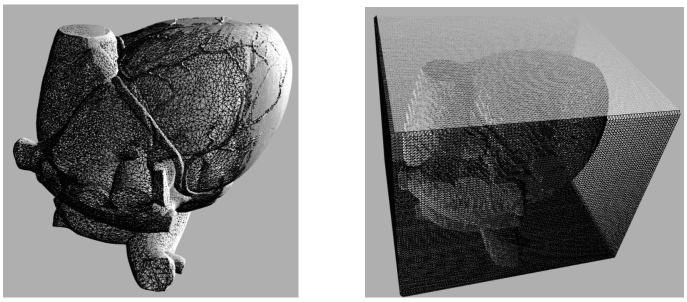
Figure 5.14: One layer of the test model after direct slicing with smooth boundary extraction.

After extracting the boundaries, CNC instructions for Arburg's droplet generating *freeformer* AM machine have been generated. For that, a contour offset distance was set on 0.28 mm (droplet diameter), and the angle of the fill paths was set on 45° (odd layers) or 135° (even layers). Based on the extracted boundary points on each of the layers, boundary contours, offset contour and fill-paths were successfully generated. One directly sliced and filled layer of the test model is depicted in Figure 5.14.

5.2.3 Inversion of models

Implementation of the Model Inversion followed the steps that were presented in Algorithm 9 in Section 4.3.3. After forming a bounding box around the voxelised model, the attenuation values have been mirrored around the central 0.5-value and transformed into a mesh.

The results of the algorithm were demonstrated on a 3D anatomical model of a human heart, see Figure 5.15a. This model is taken from *Thingiverse* database, origin: `https://www.thingiverse.com/thing:45830` from Drew Morgan, licensed under Creative Commons - Attribution - Share Alike license (CC BY-SA 3.0). Since the model was only available as a mesh data, prior to inversion it was voxelised (200^3 voxels), as described in Section 4.2.2. The resulting inverted model can be seen in Figure 5.15b. The area outside the bounding box and inside the heart is empty, while the area between the bounding box and heart borders got "filled".



(a) Original model. **(b)** Inverted model.

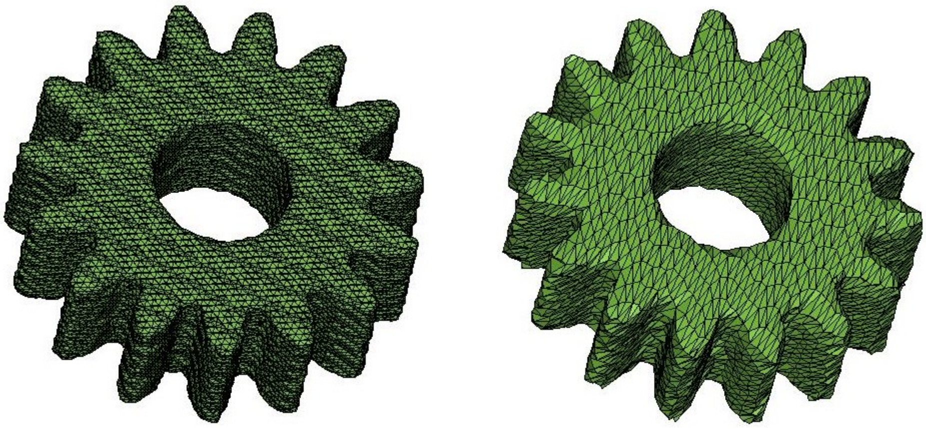
Figure 5.15: Inversion of an anatomical model of a human heart (both images are licensed under CC BY-SA 3.0 license).

5.2.4 Mesh simplification

Mesh simplification Algorithm 10 eliminates the mesh vertices which are too close to each other according to the predefined precision limits in all three directions. For that, Make-Set, Find and Union functions of the disjoint-set data structure have been implemented. The structure has then been used to merge the vertices by weighting, as introduced in Section 4.3.4.

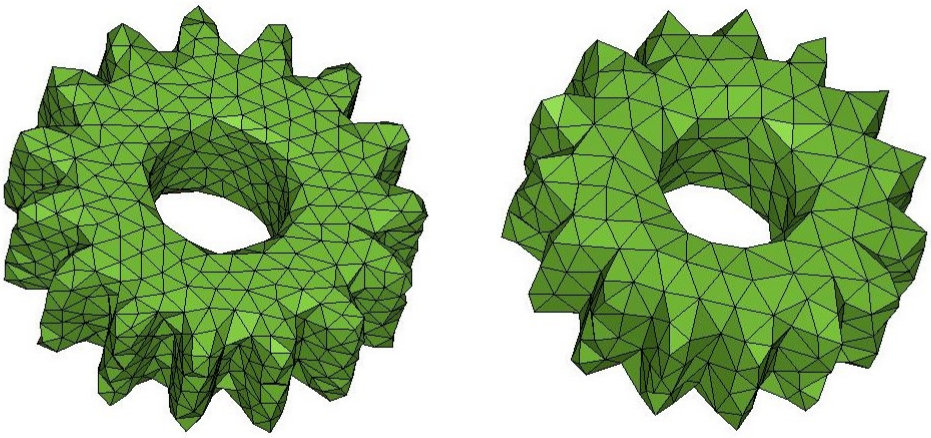
The effects of the mesh simplification are demonstrated on a gear wheel mesh model, see Figure 5.16a. The maximum length of the model is 17.25 mm and the height is 6 mm. The simplifications were successfully performed with three different limits specifications: $xLim = 0.1\,\text{mm}$, $yLim = zLim = 0.5\,\text{mm}$; $xLim = yLim = zLim = 0.5\,\text{mm}$ and $xLim = yLim = zLim = 0.7\,\text{mm}$. The simplification results can be seen in Figures

5.16b, 5.16c and 5.16d. In all three cases, the computation took at maximum 3 iterations and lasted less than 8 s on the machine introduced at the beginning of the chapter.



(a) Model prior to simplification. Vertices: 28444, Faces: 56888.

(b) Simplification with: $xLim = 0.1\,\text{mm}$, $yLim = zLim = 0.5\,\text{mm}$. Vertices: 3443, Faces: 6886, $d_H = 0.40\,\text{mm}$.



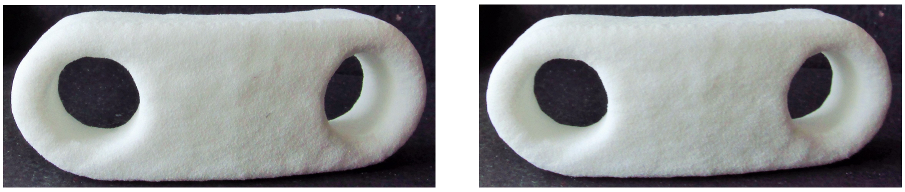
(c) Simplification with: $xLim = yLim = zLim = 0.5\,\text{mm}$. Vertices: 934, Faces: 1868, $d_H = 0.63\,\text{mm}$.

(d) Simplification with: $xLim = yLim = zLim = 0.7\,\text{mm}$. Vertices: 427, Faces: 854, $d_H = 0.91\,\text{mm}$.

Figure 5.16: Results of Mesh simplification with different limits. With higher limits, the number of vertices and faces decreases and the corresponding Hausdorff distance d_H increases.

As an error metric, the Hausdorff distance d_H was measured between the original and all the simplified meshes (performed in *MeshLab*). The original mesh has 28444 vertices. After applying the simplification with $xLim = 0.1\,\text{mm}$, $yLim = zLim = 0.5\,\text{mm}$, the number of vertices dropped down to 3443, i.e. 12% of the initial number of vertices remained. Nevertheless, the geometrical fidelity was preserved with $d_H = 0.4\,\text{mm}$. In case of $xLim = yLim = zLim = 0.5\,\mathrm{mm}$ (0.7 mm), the number of vertices dropped down to $1868, 6.5\%$ $(854, 3\%)$ with $d_H = 0.63\,\mathrm{mm}$ (0.91 mm).

The mesh simplification was additionally applied on a real-world example: after reconstructing a kinematic link with FDK algorithm from real X-ray projections (depicted in Figure 5.2), a corresponding mesh was obtained. The mesh was fabricated with *EOS Formiga P 100* 3D printer, whose resolution is limited by grain size: 0.1 mm in z-direction and 0.05 mm in x- and y-direction. One 3D object was printed without applying mesh simplification and the other one with applying mesh simplification, with limits adjusted on the corresponding printer resolution. The two printed parts are shown in Figure 5.17. The simplification with $xLim = yLim = 0.05\,\mathrm{mm}$ and $zLim = 0.1\,\mathrm{mm}$ resulted in reduction of vertices from 20800 to 17870 (86%) with $d_H = 0.03\,\mathrm{mm}$. No visual difference is noticeable.



(a) Reconstrcted 3D object without simplification.

(b) Reconstrcted 3D object with simplification.

Figure 5.17: Comparison between two 3D printed parts: one without applying mesh simplification and the other one with applying mesh simplification, with limits adjusted to the *EOS Formiga P 100* 3D printer resolution – No visual difference.

5.3 Reduction of voids

5.3.1 General geometries

The realisation of detection and filling of general voids were published by Prša et al. in July 2015 at the International Conference on Computer, Information and Telecommunication Systems (CITS), [Prša et al., 2015b].

The method proposed in Section 4.4.1 was implemented in Matlab 2014a. Prior to printing, a simulation of the void filling process with different threshold values was conducted. Afterwards, two test-parts were printed and compared: one only filled according to the contour strategy and the other one with the additional filling of the voids according to the previously proposed process. The mesh representation of a gear wheel served as a test model, see Figure 5.18.

The simulation was run with several different configurations, and the impact of different overlapping threshold values was observed. Three different filling strategies were applied: raster, contours and dense. The visual effect of the overlapping variations from 10% to 40% droplets overlapping for the test model can be observed in Figure 5.19.

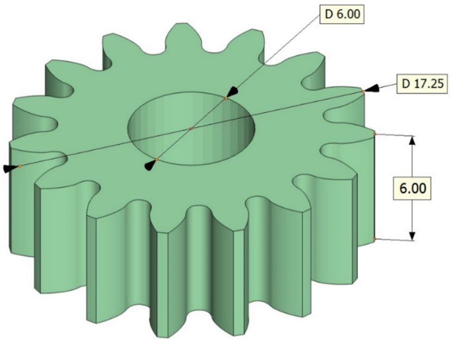
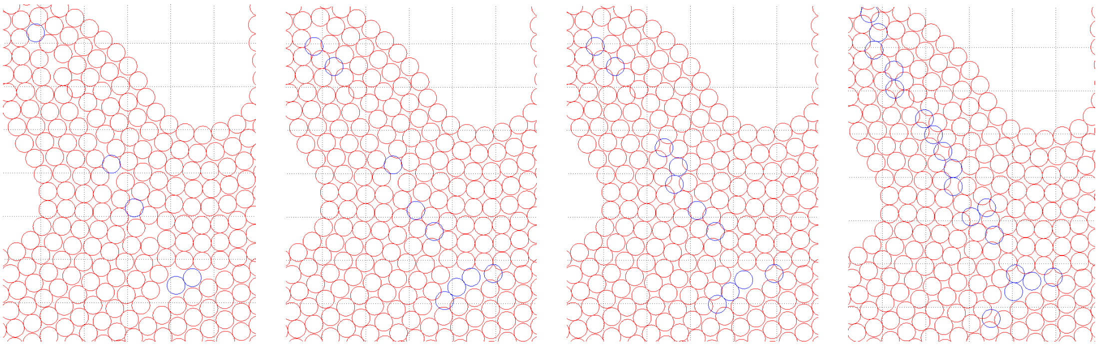


Figure 5.18: Test model with its main dimensions in *mm*.



(a) 10% Overlapping. **(b)** 20% Overlapping. **(c)** 30% Overlapping. **(d)** 40% Overlapping.

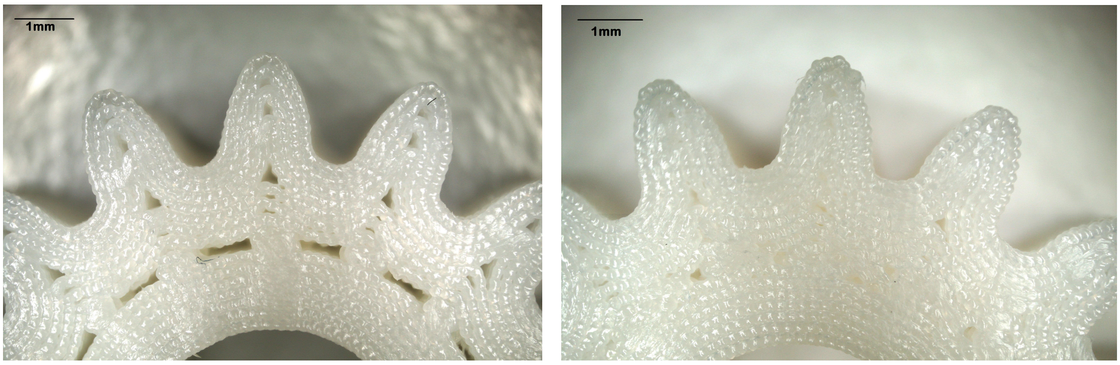
Figure 5.19: Simulation with different overlapping thresholds.

Once the gear wheel had been sliced, contour filling was applied. Two sets of test parts were built by the Arburg's droplet generating freeformer AM machine with material BASF Terluran GP-35: without and with the applied detect-and-fill-the-voids method with 20% overlapping threshold. Since two droplet central points are distant for at least two radiuses, each droplet is surrounded by some empty space due to the geometrical constraints. Within a square of side-size of two droplet radiuses, there is $1 - \frac{r^2\pi}{4r^2} = 21\%$ of unfilled space left. The additional overfill that is smaller than the surface of the surrounding voids will be consumed by the voids and will not impose additional material humps. For this reason, the overlapping threshold of 20% was chosen.

The visual comparison of the parts is shown in the Figure 5.20, which clearly shows the improvements regarding the voids reduction in case of applying the proposed method. The experiments that examine the surface roughness and fracture toughness of the printed sets will be presented in the Section 6.3.

5.3.2 Pointed ends

The realisation of a method for enhanced filling of pointed ends was published at the ASME International Mechanical Engineering Congress & Exposition (IMECE) by Prša



(a) No additional void filling applied. **(b)** Additional void filling applied.

Figure 5.20: Comparison of the printed parts.

et al. in November 2014, [Prša et al., 2014a] and at the IEEE International Conference on Robotics and Biomimetics (ROBIO) by Prša et al. in December 2014 [Prša et al., 2014b].

The algorithm proposed in Section 4.4.2 was successfully implemented. The necessary numbers of insertions and deletions for 60 angles ranging from 1° up to 60° were obtained. In the Tables 5.3 and 5.4 the under-fill, over-fill and overall errors at the initial and the final stage are shown. The last two columns indicate the exact number of droplets that are to be added and/or removed to/from the sharp angle. With decreasing of the angle, the impact of the algorithm increases. The overall error declined for up to 30%.

Two 3D models shaped as triangular prisms were created in order to demonstrate the effectiveness of the additional droplet insertions and deletions. The first prism has a pointed end with 15°, and the second one with 5°. The height of the prisms (3D model) is 5 mm. Both models were printed with the Arburg's droplet generating freeformer AM machine out of Terluran GP-35 Acrylonitrile butadiene styrene (ABS) material with a slice width of 0.215 mm (approximated on altogether 23 layers with a total height of 4.945 mm).

The 15°-prism has been sliced (in the direction that is vertical to the base) and the contour filling method has been applied as the infill strategy. Two different parts were printed using the aforementioned printer: one part with the three additional insertions and one deletion as suggested by the proposed algorithm, and one part without. In the Figure 5.21a, voids and over-fills in a pointed tip are visible. Since the infill is consisted of repeating contours, those voids are repeating with each new contour level. For this reason, a wavy form of a pointed end has been induced, instead of a straight-lined pointed end defined by the geometry. Once these voids have been filled in, the wavy form diminishes, see Figure 5.21b.

Regarding the 5°-prism, five different infill strategies were applied in order to fill its interior: two contours followed by a raster infill, one contour followed by a raster infill, repetitive contours infill (algorithm not applied), repetitive contours infill with eight droplet insertions and repetitive contours infill with eight additional droplets and four

α	init. E_-	fin. E_-	init. E_+	fin. E_+	init. E	fin. E	$\#_{ins}$	$\#_{del}$
1°	0.2961	0.2287	0.0816	0.0338	0.3778	0.2625	46	23
2°	0.2950	0.2302	0.0799	0.0361	0.3750	0.2663	23	11
3°	0.2920	0.2311	0.0783	0.0381	0.3703	0.2692	15	7
4°	0.2906	0.2268	0.0766	0.0361	0.3672	0.2629	11	5
5°	0.2881	0.2341	0.0750	0.0349	0.3631	0.2691	8	4
6°	0.2896	0.2344	0.0734	0.0401	0.3630	0.2745	7	3
7°	0.2858	0.2250	0.0718	0.0316	0.3575	0.2566	6	3
8°	0.2829	0.2253	0.0702	0.0379	0.3530	0.2632	5	2
9°	0.2810	0.2201	0.0687	0.0399	0.3497	0.2600	5	2
10°	0.2825	0.2234	0.0672	0.0300	0.3496	0.2534	4	2
11°	0.2799	0.2183	0.0655	0.0296	0.3454	0.2479	4	2
12°	0.2786	0.2240	0.0642	0.0395	0.3428	0.2635	3	1
13°	0.2731	0.2207	0.0626	0.0394	0.3357	0.2602	3	1
14°	0.2721	0.2190	0.0609	0.0446	0.3330	0.2635	3	1
15°	0.2743	0.2137	0.0597	0.0414	0.3340	0.2551	3	1
16°	0.2781	0.2347	0.0584	0.0225	0.3365	0.2572	2	1
17°	0.2675	0.2189	0.0568	0.0278	0.3243	0.2467	2	1
18°	0.2659	0.2148	0.0551	0.0261	0.3210	0.2409	2	1
19°	0.2690	0.2226	0.0535	0.0291	0.3224	0.2517	2	1
20°	0.2705	0.2171	0.0526	0.0260	0.3231	0.2431	2	1
21°	0.2686	0.2121	0.0516	0.0261	0.3202	0.2382	2	1
22°	0.2723	0.2209	0.0503	0.0285	0.3227	0.2494	2	1
23°	0.2570	0.2252	0.0489	0.0085	0.3059	0.2337	1	1
24°	0.2560	0.2206	0.0473	0.0473	0.3033	0.2680	1	0
25°	0.2601	0.2205	0.0457	0.0457	0.3057	0.2661	1	0
26°	0.2546	0.2152	0.0439	0.0439	0.2985	0.2591	1	0
27°	0.2497	0.2130	0.0423	0.0423	0.2919	0.2553	1	0
28°	0.2547	0.2192	0.0407	0.0407	0.2955	0.2599	1	0
29°	0.2609	0.2242	0.0396	0.0396	0.3005	0.2638	1	0
30°	0.2623	0.2226	0.0391	0.0391	0.3015	0.2617	1	0

Table 5.3: Algorithm results before and after applying the algorithm (insertion and deletion). To each angle from **1°** to **30°**, a number of additional droplets to be inserted and a number of superfluous droplets to be removed is assigned.

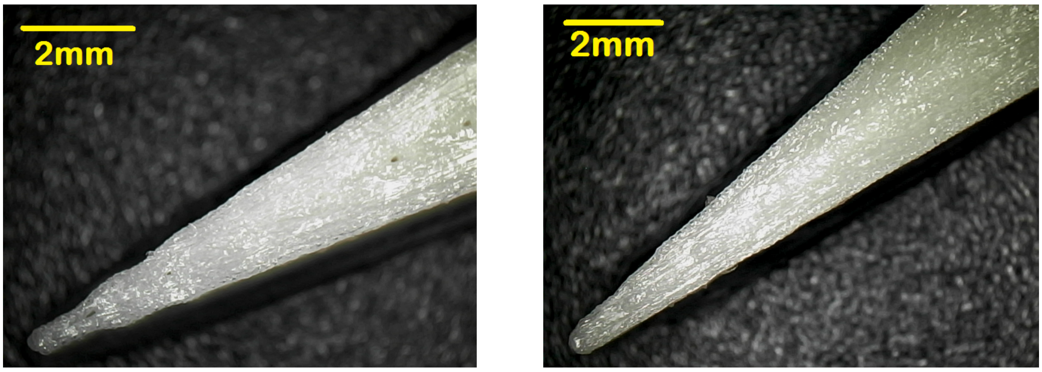
deletions of superfluous droplets. To ensure appropriate material bonding, it is important to mention that there exists a certain overlap between the contour and the raster. The raster boundary has been offset for a droplet radius, instead of a droplet diameter, from the inner most contour.

The height of the five parts has been measured at three different positions between the very tip (first contour) and the second contour (raster): tip itself, the lowest point and

α	init. E_-	fin. E_-	init. E_+	fin. E_+	init. E	fin. E	$\#_{ins}$	$\#_{del}$
31°	0.2533	0.2094	0.0385	0.0385	0.2917	0.2479	1	0
32°	0.2527	0.2074	0.0378	0.0488	0.2905	0.2562	1	0
33°	0.2563	0.2107	0.0370	0.0495	0.2932	0.2602	1	0
34°	0.2602	0.2153	0.0361	0.0512	0.2963	0.2666	1	0
35°	0.2618	0.2196	0.0351	0.0548	0.2969	0.2744	1	0
36°	0.2585	0.2184	0.0340	0.0577	0.2925	0.2761	1	0
37°	0.2451	0.2060	0.0329	0.0329	0.2779	0.2389	1	0
38°	0.2347	0.1955	0.0316	0.0601	0.2663	0.2556	1	0
39°	0.2313	0.1910	0.0303	0.0595	0.2617	0.2505	1	0
40°	0.2315	0.1888	0.0290	0.0578	0.2605	0.2466	1	0
41°	0.2333	0.1878	0.0276	0.0555	0.2609	0.2433	1	0
42°	0.2357	0.1893	0.0261	0.0550	0.2618	0.2443	1	0
43°	0.2376	0.1909	0.0246	0.0552	0.2622	0.2460	1	0
44°	0.2383	0.1912	0.0230	0.0553	0.2613	0.2465	1	0
45°	0.2366	0.1893	0.0214	0.0555	0.2579	0.2447	1	0
46°	0.2310	0.1836	0.0198	0.0557	0.2507	0.2392	1	0
47°	0.2208	0.1734	0.0181	0.0561	0.2389	0.2295	1	0
48°	0.2160	0.1688	0.0164	0.0566	0.2325	0.2254	1	0
49°	0.2145	0.1677	0.0147	0.0575	0.2292	0.2252	1	0
50°	0.2149	0.1690	0.0131	0.0588	0.2280	0.2278	1	0
51°	0.2166	0.2166	0.0114	0.0114	0.2280	0.2280	0	0
52°	0.2192	0.2192	0.0098	0.0098	0.2289	0.2289	0	0
53°	0.2222	0.2222	0.0082	0.0082	0.2303	0.2303	0	0
54°	0.2253	0.2253	0.0066	0.0066	0.2319	0.2319	0	0
55°	0.2283	0.2283	0.0051	0.0051	0.2334	0.2334	0	0
56°	0.2309	0.2309	0.0037	0.0037	0.2346	0.2346	0	0
57°	0.2329	0.2329	0.0025	0.0025	0.2354	0.2354	0	0
58°	0.2341	0.2341	0.0014	0.0014	0.2355	0.2355	0	0
59°	0.2343	0.2343	0.0000	0.0000	0.2348	0.2348	0	0
60°	0.2332	0.2332	0.0000	0.0000	0.2332	0.2332	0	0

Table 5.4: Algorithm results before and after applying the algorithm (insertion and deletion). To each angle from **31°** to **60°**, a number of additional droplets to be inserted and a number of superfluous droplets to be removed is assigned.

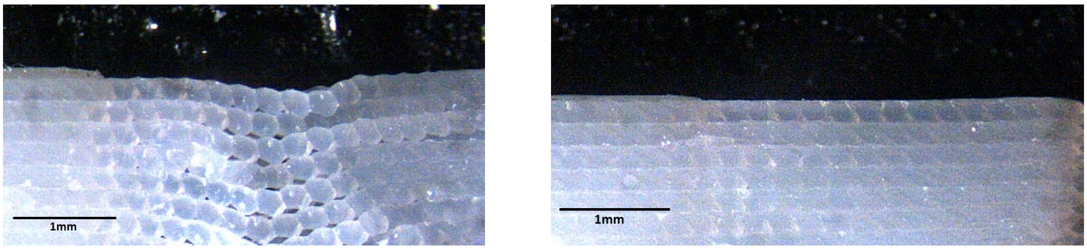
the highest point. The biggest height oscillations (588 µm) are present for the repetitive contours infill, followed by the two contours and a raster infill (336 µm) and one contour and a raster infill (170 µm). After the insertion of additional eight droplets, the height oscillations decreased to 138 µm. At last, after eight insertions and four deletions, the corresponding upper surface of the printed part resulted in the height oscillation of only 97 µm.



(a) Algorithm not applied. **(b)** Algorithm applied.

Figure 5.21: Comparison of two printed 15° pointed ends: with and without applying the proposed algorithm.

The comparison between the pointed end with plane repetitive contours infill strategy and the enhanced one can be seen in Figure 5.22. Without taking in account filling of the voids while contour printing, the voids on each level are accumulating and lead to decaying of the whole structure. Moreover, the accumulated induced overlappings that appear directly after the voids lead to the more uneven final surface of the part. Since the over-fills and under-fills were minimised by utilising the suggested algorithm, the top surface is more even.



(a) Plane repetitive contours infill strategy. **(b)** Improved repetitive contours infill strategy according to the algorithm.

Figure 5.22: Zoomed side view on 5° pointed tip printed with and without applying of the algorithm.

6 Experiments and Analysis

This chapter presents the empirical results of the previously introduced components. As a proof of concept, a 3D replication of a reference object will be presented in Section 6.1. Measuring of the reconstruction computational times and the reconstruction errors follow in Section 6.2. Finally, in Section 6.3, a mechanical evaluation of the AM parts will be presented.

6.1 Proof of concept

To demonstrate a proof of concept of the replication cycle (shown in Figure 4.1), a reference 3D object was replicated. The chosen reference object is a cube with a cylindrical bore ($20 \times 20 \times 20$ mm, $r = 5$ mm), made of fine polyamide 12 (PA 2200), depicted in Figure 6.1. *Leybold X-ray apparatus 554 800* (LD DIDACTIC) was used for scanning. Regarding the fabrication, three different AM machines were used: EOS *Formiga P 100* (Selective Laser Sintering) with polyamide 12 - PA 2200, Stratasys *Objet30 Prime* (PolyJet) with rigid photopolymer RGD525, and ARBURG *freeformer* (ARBURG Plastic Freeforming) with BASF Terluran GP-35 ABS.

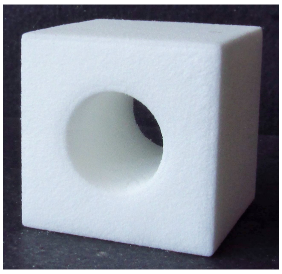


Figure 6.1: Original reference object: cube with a cylindrical bore ($20 \times 20 \times 20$ mm, $r = 5$ mm) out of polyamide 12 (PA 2200).

The 3D replication consisted of following steps:

1) Fetch the *X-ray tube parameters* from the database for the specific object.

2) After setting up the X-ray tube parameters within an X-ray apparatus, obtain *X-ray projections* of the object.

3) Perform *3D reconstruction* based on the projections.

4) *Correct planes* on the reconstructed 3D model.

For further steps choose either path a) or path b):

5a) Transform the corrected voxelised model into a *mesh*.

6a) *Simplify* the *mesh* according to the resolution of the AM machine.

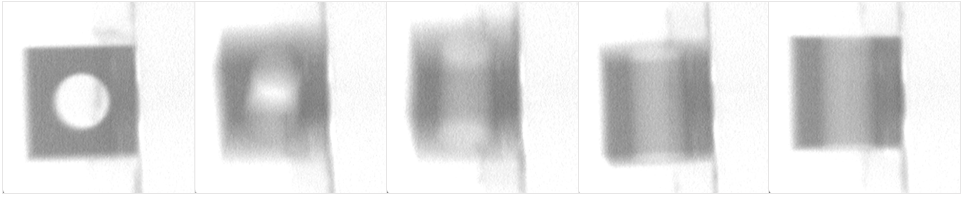
7a) *Fabricate* simplified mesh.

or:

5b) Perform *direct slicing* on the corrected voxelised model according to the AM machine specifications.

6b) *Fabricate* the part with the resulting machine instructions.

The utilised *Leybold X-ray apparatus* provides maximally 35 kV voltage. According to the database in Table 5.2 and obtained limits for PA material in Figure 5.5a (Section 5.1.2), scanning of the reference object with 35 kV demands at least 400 µA. After setting the parameters and fixing the object with the holders of lower material density, two sets of 180 X-ray projections (250 × 250 pixels) were made. The first set contains empty projections, i.e. one projection for each 2° without the object (see Figure 6.2b). The second set contains projections with the object, as shown in Figure 6.2a. The total distance between the X-ray source and the detector was 397 mm, and the object was positioned in a way that the distance between the X-ray source and the object was 187 mm.

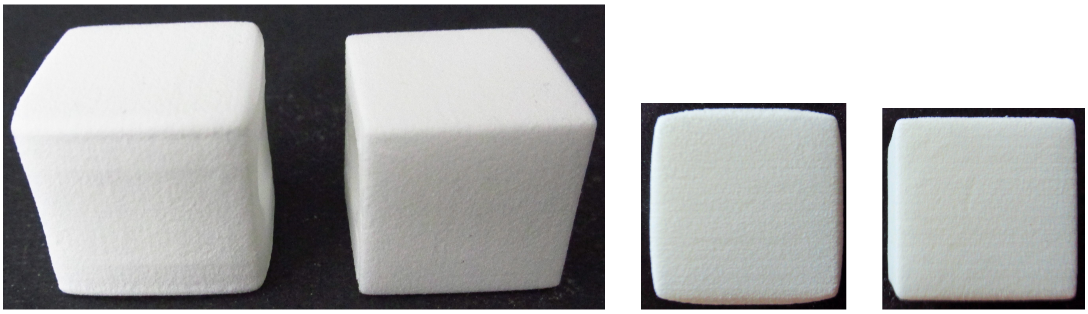


(a) X-ray projections.

(b) Empty X-ray projections

Figure 6.2: Several X-ray projections and empty projections of the reference object (corresponding pairs are horizontally aligned). Light grey shadows in the background appear due to the holders that stabilise the object.

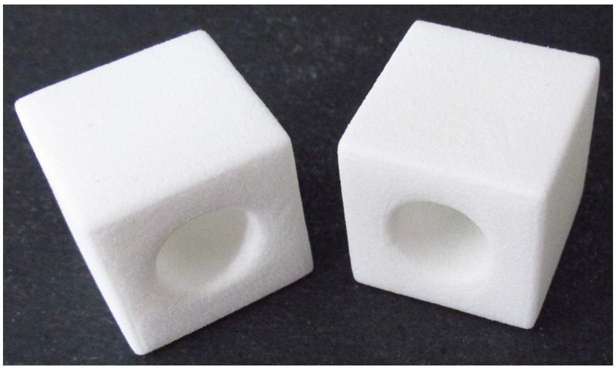
Based on the obtained projections, a 3D reconstruction was conducted with 7 iterations of SART with 128^3 voxels of 0.26^3 mm^3 size. The computational time for the reconstruction process was 5.16 min on the machine introduced at the beginning of Chapter 5. Before applying any corrections, the raw reconstructed model was printed with *Formiga P 100*, *Objet30 Prime* and *freeformer*. The results are shown in Figures 6.3a, 6.4a and 6.5a respectively.



(a) 3D replica without any corrections (left) and with applied plane corrections (right).

(b) No corrections (top view).

(c) With corrections (top view).

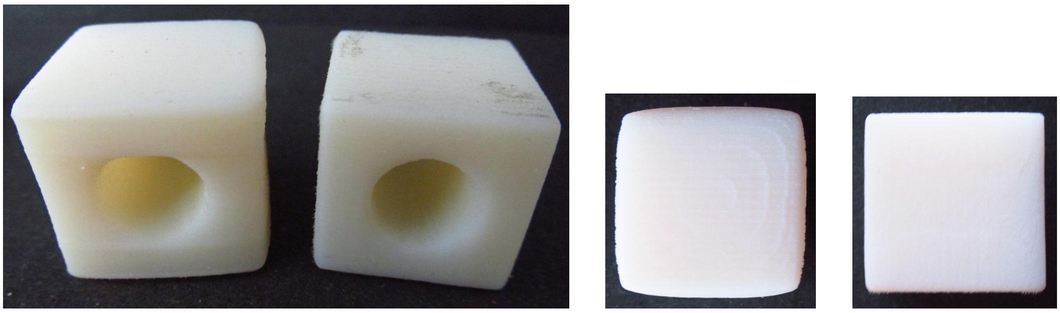


(d) 3D corrected replica without mesh simplification (left) and with applied mesh simplification (right): no impact on the final part.

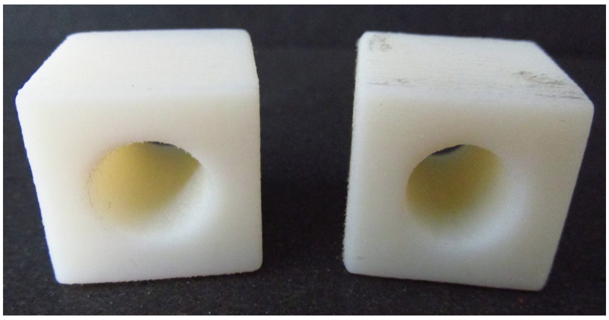
Figure 6.3: Realisation of the 3D replicas with *Formiga P 100*.

After obtaining a reconstructed voxelised model, planes' corrections were conducted with 10 iterations of Smearing Correction. The computational time for the plane detection process of all six planes on a converted PC with 46666 points was 21.1 s and for the corresponding voxel correction 3.5 s (on the previously introduced machine). The corrected model was printed with the above mentioned AM machines. The results are shown in Figures 6.3, 6.4 and 6.5 a-c. The direct visual comparison of the corrected and raw replicas indicates that the corrected objects were significantly enhanced. The rounded structure of the raw objects diminished, and sharp corners appeared. The resulting corrected replicas are more resembling the original reference object (depicted in Figure 6.1).

In addition to the visual comparison, a geometrical comparison between the raw mesh and the corrected mesh was conducted. The authors in [Prša et al., 2015a] proposed a software tool for the automatic conformity evaluation of mesh models. The evaluation tool includes four conformity tests: *flatness* of the planes, *orthogonality* of perpendicular planes, *roundness* of the circular portions, and *coaxiality* of the whole model. Each test results in a deviation value from the optimal score (0). The results of the mesh-evaluation



(a) 3D replica without any corrections (left) and with applied plane corrections (right). **(b)** No corrections (top view). **(c)** With corrections (top view).



(d) 3D corrected replica without mesh simplification (right) and with applied mesh simplification (left): no impact on the final part.

Figure 6.4: Realisation of the 3D replicas with *Object30 Prime* (Additional note: black stains on the surface are common for this type of manufacturing and appear independently of the object quality.)

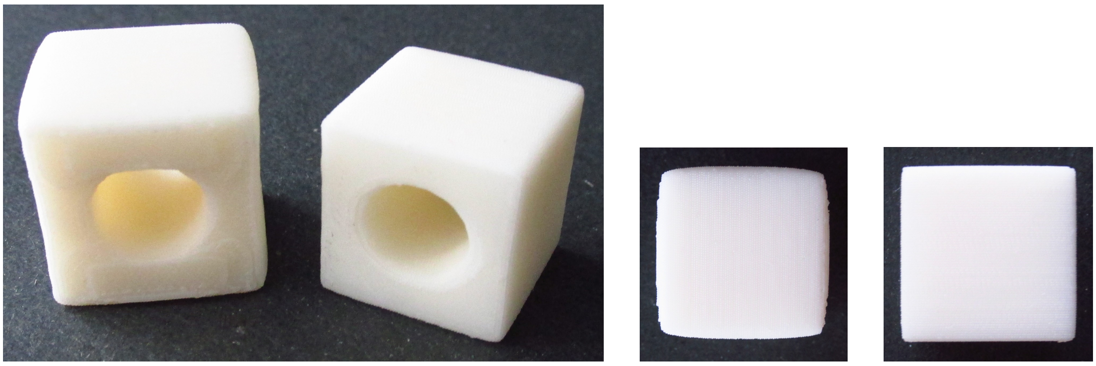
are presented in Table 6.1, which clearly shows that the corrected mesh is superior to the raw mesh.

Table 6.1: Mesh-evaluation results: geometrical comparison between the raw mesh and the corrected mesh.

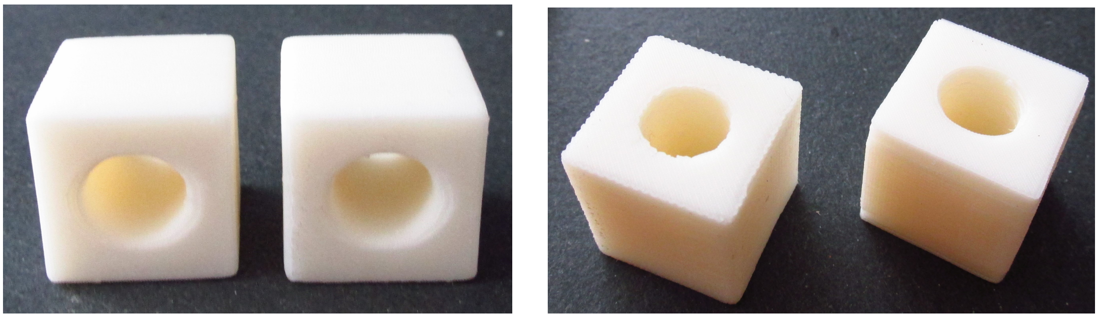
Conformity test	Results for raw mesh	Results for corrected mesh
flatness	0.59 mm	0.13 mm
orthogonality	9.32°	3.12°
roundness	0.75 mm	0.58 mm
coaxiality	3.30 mm	1.20 mm

As stated before, the corrected voxelised model can be either directly sliced or transformed into mesh. Regarding the latter approach, mesh simplification was conducted. Here, the resolution limitations of the aforementioned AM machines were considered:

Formiga P 100 is in the printing plane limited by the material grain size with $xLim = yLim = 0.05\,\text{mm}$ and in the slicing axis with $zLim = 0.1\,\text{mm}$.



(a) 3D replica without any corrections (left) and with applied plane corrections (right). **(b)** No corrections (top view). **(c)** With corrections (top view).



(d) 3D corrected replica without mesh simplification(left) and with applied mesh simplification (right): no impact on the final part. **(e)** 3D replicas realised with direct slicing: binary approach (left) and smooth approach (right).

Figure 6.5: Realisation of the 3D replicas with *freeformer*.

Object30 Prime is in the printing plane limited by the plotter resolution with $xLim = yLim = 0.042\,\text{mm}$ and in the slicing axis with $zLim = 0.016\,\text{mm}$.

freeformer is in all three directions limited by the droplet diameter, which slightly decreases in the slicing direction: $xLim = yLim = 0.281\,\text{mm}$ and $zLim = 0.27\,\text{mm}$.

With these limitations, the initial mesh with 40439 vertices was simplified. Regarding the limitations of *Formiga P 100*, the number of vertices dropped down to 38044 (94%) with the Hausdorff distance $d_H = 0.024\,\text{mm}$. The computation lasted for 9.5 s on the previously introduced machine. The limitations of *Object30 Prime* led to 39011 vertices (96.5%) with $d_H = 0.014\,\text{mm}$ (9.8 s). In the case of *freeformer*, the number of vertices dropped down to 9085 (22.5%) with $d_H = 0.194\,\text{mm}$ (10.5 s). The simplified meshes have been printed and compared against the corrected meshes that were not simplified. The results can be seen in Figures 6.3d, 6.4d and 6.5d respectively. The visual comparison shows that the mesh simplification according to the printers' resolution limitations does not have any impact on the final object quality.

If there is an access to the AM machine's readable code, direct slicing can be performed. The available knowledge of the *freeformer's* G-code format was utilised, and direct slicing of the corrected voxelised model was carried out. The computational time of the total boundary extraction was below 1 s. The generation of the machine-interpretable instructions lasted around 1 min. Regarding the boundary extraction, both binary and smooth approaches have been applied. The final results of the fabricated objects are shown in Figure 6.5e. Both replicas are shape-preserving, even though the smooth approach leads to better results.

With all these steps, a full replication cycle for one reference object was successfully conducted.

6.2 Reconstruction unit

This section presents the influence of different reconstruction parameters on the reconstruction quality and computational time. All measurements in this section were carried out on a 64bit PC with Intel i7-6700HQ CPU, 16GB RAM and NVIDIA GeForce GTX 960M graphics card using Linux.

A 3D mesh model of a cube with a cylindrical bore ($20 \times 20 \times 20$ mm, $r = 5$ mm) served again as a test model, see Figure 5.3a. After voxelisation in 512^3 voxels of 0.065^3 mm^3 size, 720 virtual projections with 300×300 pixels were made with the X-ray simulator. Based on these projections, 3D reconstructions with 64^3 (0.523^3 mm^3), 128^3 (0.261^3 mm^3) and 256^3 (0.131^3 mm^3) voxels were calculated and the corresponding computational time was measured. The obtained voxelised 3D models were transformed into 3D meshes. The Hausdorff distances d_H between these meshes and the original 3D mesh model served as a measure for the reconstruction quality. The distances were measured by using the software tool *Metro*, [Cignoni et al., 1998].

The following diagrams in Figures 6.6, 6.8 and 6.9 show several measurements for different parameter configurations as points, with a measured running time and calculated error as coordinate axes. The advantage of this representation is a fast revealing of the optimal parameter-sets. All the points that are not positioned in the lower left convex hull of all measurement-points, do not have perfect parameter settings. They are outperformed by other settings that yield the same error but with a lower running time, or by other settings that yield a lower error at the cost of the same running time.

The first set of measurements was done with the *FDK* reconstruction algorithm. The number of projections (Proj) varied between 45, 90, 180, 360 and 720. The diagram with the results of successfully reconstructed models is shown in Figure 6.6.

The FDK reconstruction based on 45 projections did not lead to reasonable results, see Figure 6.7a. From 90 projections onwards the reconstructions were reasonable, see Figure 6.7b. The computational time increased with each increment of the projection number, but the error, however, did not significantly sink (for the 128^3 voxel-series it even surprisingly increased). This implies that there exists a threshold for a minimal number of projections that are necessary to obtain reasonable reconstruction results, and it does not pay off to use more projections than the threshold. Regarding the voxel number, their incremental leads, as expected, to longer computational times and smaller errors. The achieved errors are in general satisfyingly small, especially for the models

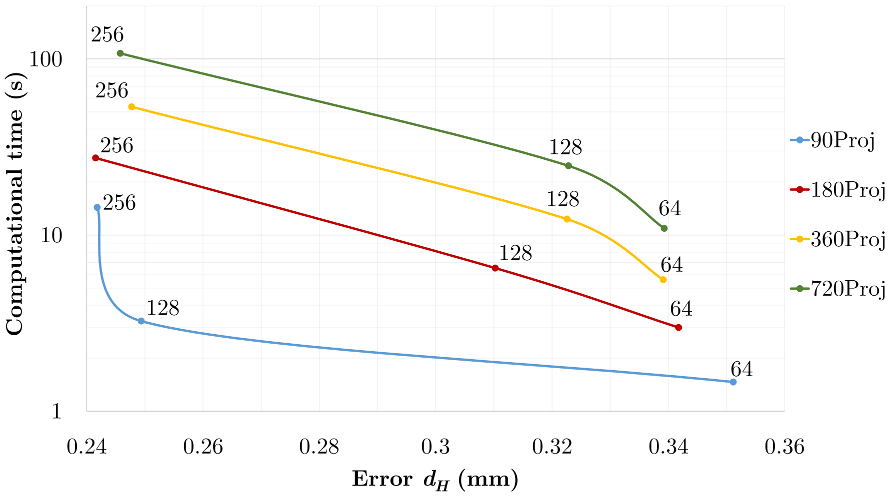
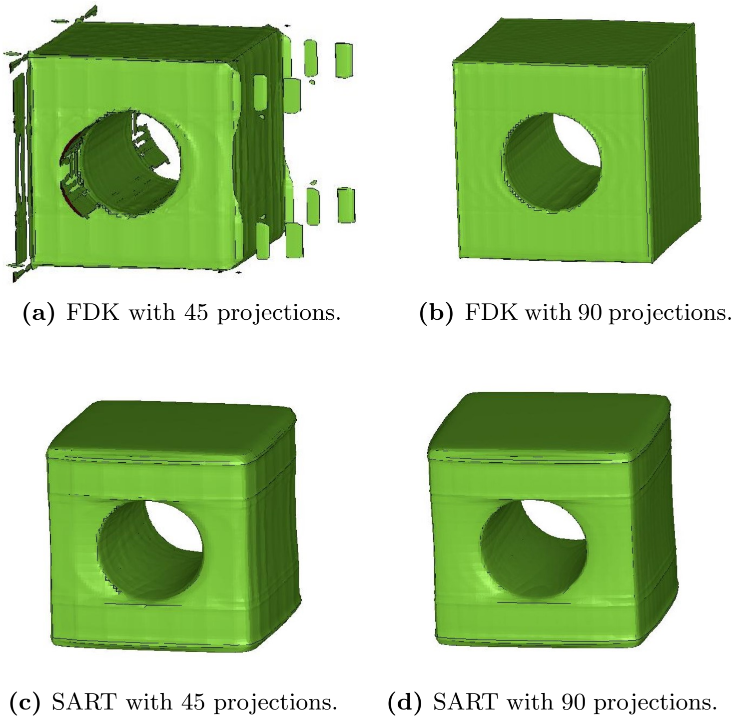


Figure 6.6: Comparison of reconstruction results with FDK. Each of the points in the diagram is assigned to the underlying voxel number in one direction.

with 64^3 voxels of 0.523^3 mm^3 size. For these models, the errors were smaller than 0.36 mm, which is below the underlying voxel resolution.



(a) FDK with 45 projections. **(b)** FDK with 90 projections.

(c) SART with 45 projections. **(d)** SART with 90 projections.

Figure 6.7: Resulting mesh models from FDK and SART (5It) reconstructions with 128^3 voxels.

The second set of measurements was done with the *SART* reconstruction algorithm. The number of iterations (It) varied between 3, 5 and 7 and the number of projections (Proj) varied between 20, 45, 90, 180, 360 and 720. The diagram with the results of the reconstructed models is shown in Figure 6.8.

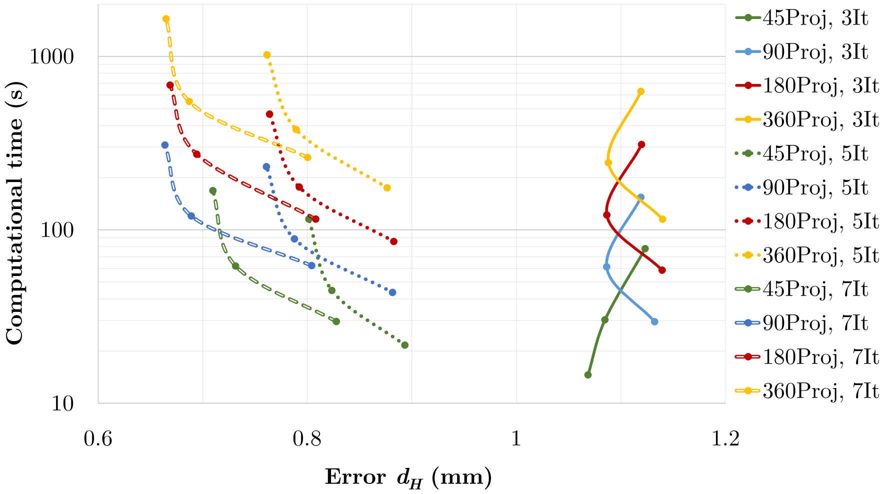


Figure 6.8: Comparison of reconstruction results with SART. Each of the points in the diagram is assigned to the underlying voxel number in one direction: 256 (highest points along each line), 128 (middle points) and 64 (lowest points).

The number of iterations had the most significant impact on the quality of reconstruction results. The models with 3 iterations were in general so imprecise that the incremental of other parameters did not lead to reasonable effects. The models with 5 and 7 iterations led to increasingly more precise models, but at the cost of higher computational times. The same can be said for the incremental of the voxel numbers.

Different from the FDK results, reasonable SART results were also obtained with a low number of projections (e.g. 20 or 45), see Figure 6.7c. However, concerning the computational time for the obtained error, 90 projections seem to be the best choice, as depicted in Figure 6.7d. A higher number of projections lead to linear time increment without a significant gain in model accuracy.

A direct comparison between FDK and SART is presented in Figure 6.9. For most parameter settings, FDK is superior to SART. FDK reconstructions were always significantly faster and mostly of better quality. SART, however, enables reconstruction for a small number of available projections, which is the case if the obtaining of projections is time-costly or the radiation dose has to be limited.

In order to measure the benefits of the parallelised computations on GPU, a final set of measurements was conducted with FDK and SART (7 iterations) with 90 projections. The resulting diagram is presented in Figure 6.10. For both SART and FDK, the computational time significantly reduced with using GPU. FDK computational times with GPU are halved for each voxel number when compared to computational times with only CPU. Regarding SART, the benefits of the parallelisation increase with the increment

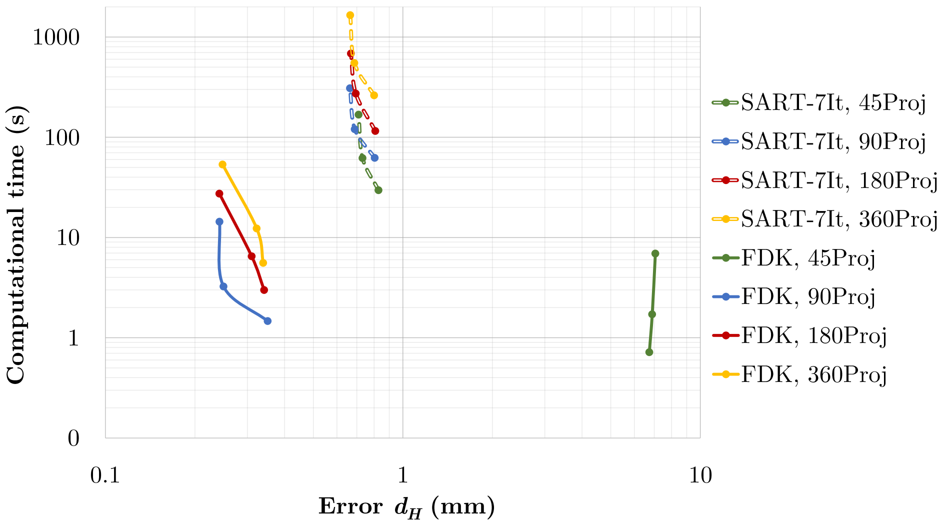


Figure 6.9: A direct comparison between FDK and SART reconstruction results. Each of the points in the diagram is assigned to the underlying voxel number in one direction: 256 (highest points along each line), 128 (middle points) and 64 (lowest points).

of the voxel numbers. For 512^3 voxels, SART computational time with using GPU is half of the corresponding computational time without using GPU.

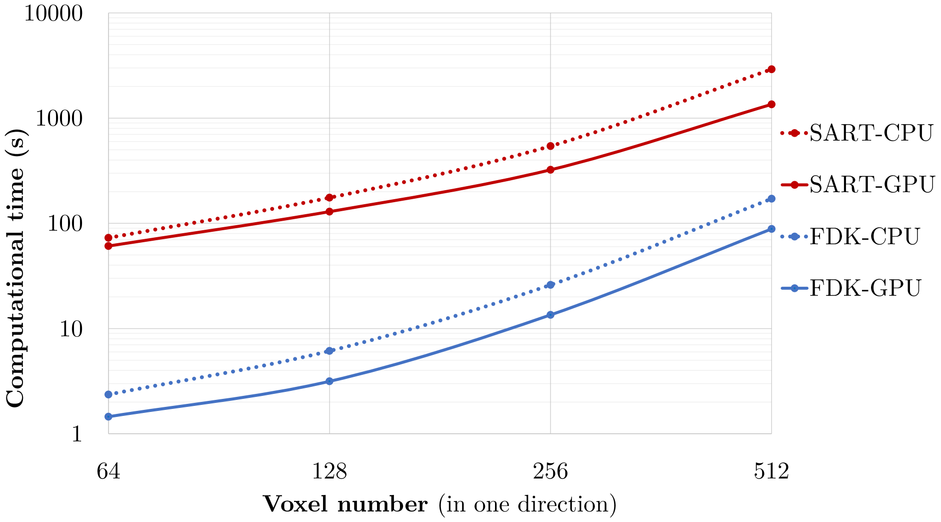


Figure 6.10: Comparison between the computational times on GPU and CPU.

6.3 Mechanical evaluation of enhanced AM parts

The mechanical evaluation of the enhanced AM parts, by the means of voids reduction in general geometries, was published by Prša et al. in July 2015 at the International Conference on Computer, Information and Telecommunication Systems (CITS), [Prša et al., 2015b].

In order to examine the impact of the method for voids reduction (proposed in 4.4.1), a test model was used as introduced in Section 5.3.1. Two sets of nine parts each were built by the Arburg's droplet generating freeformer machine for AM with material BASF Terluran GP-35: without and with applied detect-and-fill-the-voids method with 20% overlapping threshold. Two mechanical properties were analysed: *surface roughness* and *fracture toughness*.

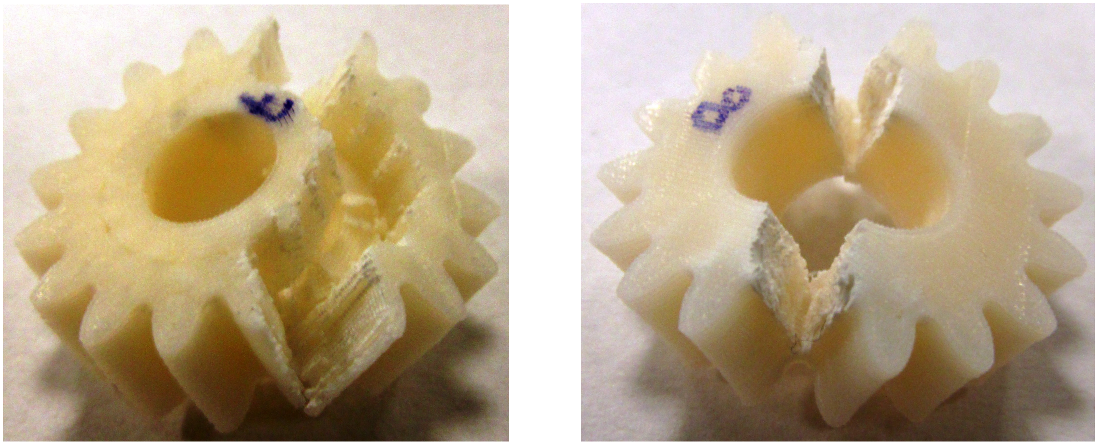
Surface roughness

In this experiment, surface roughness was measured on the top slice of each part at two randomly chosen positions. The experiment was carried out using the surface roughness measuring device Mahr Perthometer M2, which consists of a sensitive needle that measures average height displacements along 5 mm long path. The first unenriched set of parts resulted in average surface roughness of 5.88 µm, while the enriched set resulted in 4.64 µm. A t-test was applied to two sets in order to obtain a statistical comparison. The enriched parts resulted in the average surface roughness reduction of 21% with 99.5% of significance.

Fracture toughness

In order to test the firmness of the parts, in the middle of each individual part (out of two sets) force was increasingly applied until the fracture occurred. Parts were placed at the centre on a simple lever press, where a three point flexural test with 5 kN sensor and Digiforce 9310 software was performed. This hand-driven machine has force and displacement sensors and with manually applied force generates stress-displacement curves. From these curves, a breaking point could be easily interpreted. The first unenriched set of parts resulted in average fracture toughness of 0.69 kN, while the enriched set resulted in 0.74 kN. The enriched parts resulted in the average fracture toughness enhancement of 7% with 99% of significance.

In the Figure 5.20 (a) it is clear that the voids are accumulated along the encountering points between the outer and inner contours. After force being applied on the parts without the detect-and-fill-voids method, breakings arose along the accumulated voids. Once the voids were filled, fracture appeared in the middle of the parts, where the force was actually applied (Figure 6.11).



(a) No additional void filling applied: fracture along the accumulated voids.

(b) Additional void filling applied: fracture according to the medial force application.

Figure 6.11: Breaking pattern of the printed parts.

7 Summary and outlook

The presented thesis describes concepts for an automated X-ray based replication of real-world objects. Rapid prototyping environment has been used as a general setting. Within this environment, reverse engineering, model processing and Additive Manufacturing have been coupled.

Up to this date, a 3D replication with an integrated X-ray digitizing unit has not been automatised. Not only is there no 3D replicating device, but also a computational coupling of the processes involved in the replication is missing too. Therefore, the underlying processes are disconnected, and there is no adjustment of the subsequent processes.

Within the scope of this thesis, X-ray digitization, post-processing and AM have been linked by taking into account the requirements and specifications of each of the processes. The customisation of the scanning and post-processing processes on different AM procedures enabled important model distinctions and effort was invested only in relevant geometrical features.

As a part of X-ray based digitization, a cone-beam reconstruction unit, an X-ray simulator and an X-ray tube parameters database have been developed. The cone-beam reconstruction unit provides an adjustable reconstruction of 3D models from a stack of X-ray projection images. It is enhanced with an interface which enables user interaction and parameter customisation, such as reconstruction algorithm, voxel size and allowed error. The X-ray simulator enables the simulation of the virtual projections from already existing 3D models. In order to automatise the setting of X-tube parameters, a database which links relevant materials and sizes with the current and voltage of the X-ray tube was established.

Within the post-processing unit there were realised: recognition and correction of planes, direct slicing of voxel-based models, inversion of models and mesh simplification. The planes of the reconstructed voxel-based models are automatically corrected by a plane detection and correction. Transformations to either sliced AM layers, triangular meshes or inverted triangular meshes were realised. A direct slicing mode enables the segmentation of previously corrected models and translates them into machine-readable instructions for AM. Extraction of meshes was enhanced with a mesh simplification tool. Based on the printing resolution of a chosen AM system, meshes are automatically simplified in a way that no mesh vertices are closer than the printer resolution.

Regarding the AM unit, an optimisation of the infill faults within the process planning for droplet-based AM extruding systems was introduced. The realised software tool corrects various under-fill and over-fill defects after launching arbitrary fillings.

With the described units, multiple replicas of a reference object have been successfully realised. As a part of the experiments, measurements of the reconstruction time and geometrical deviations have been conducted, and impacts of different parameters on the final results have been investigated. A mechanical evaluation (surface roughness and fracture toughness) of the AM parts was carried out as well.

Within the scope of this thesis, an automated computational pipeline for 3D replication was realised. Therefore, a 3D replicating device with an X-ray based digitization is feasible. Manufacturing such a device, based on the introduced concepts, would be the next challenge. For that, an extension of the proposed X-ray database with new materials and sizes (related to the specific AM system) is necessary.

Furthermore, the automatic detection and correction unit is currently limited to the planes geometries. Detection and correction of other primitives (cylinders and spheres) and other non-primitives would enrich the correction process. While correcting the geometries, the goal of reaching the perfect geometries is uncompromisingly followed. This may lead to somewhat changed dimensions of the original object. Introducing size preservation as a parallel goal would lead to even more precise replicas.

Bibliography

[Agarwala et al., 1996] Agarwala, M. K., Jamalabad, V. R., Langrana, N. A., Safari, A., Whalen, P. J., and Danforth, S. C. (1996). Structural quality of parts processed by fused deposition. *Rapid Prototyping Journal*, 2(4):4–19.

[Amenta et al., 2001] Amenta, N., Choi, S., and Kolluri, R. K. (2001). The Power Crust. In *Proceedings of the Sixth ACM Symposium on Solid Modeling and Applications*, SMA '01, pages 249–266, New York, NY, USA. ACM.

[Andersen and Kak, 1984] Andersen, A. H. and Kak, A. C. (1984). Simultaneous algebraic reconstruction technique (SART): a superior implementation of the ART algorithm. *Ultrasonic imaging*, 6(1):81–94.

[Beraldin et al., 2000] Beraldin, J., F.Blais, Cournoyer, L., Godin, G., and Rioux, M. (2000). Active 3D Sensing - A Tutorial. In *International Conference on 3D Digital Imaging and Modeling*.

[Berger et al., 2014] Berger, M., Tagliasacchi, A., Seversky, L., Alliez, P., Levine, J., Sharf, A., and Silva, C. (2014). State of the art in surface reconstruction from point clouds. In *EUROGRAPHICS star reports*, pages 161–185.

[Bertoldi et al., 1998] Bertoldi, M., Yardimci, M. A., Pistor, C. M., and Güceri, S. I. (1998). Domain Decomposition and Space Filling Curves in Toolpath Planning and Generation. *Solid Freeform Fabrication Symposium*, pages 267–276.

[Bibb et al., 2010] Bibb, R., Eggbeer, D., and Evans, P. (2010). Rapid prototyping technologies in soft tissue facial prosthetics: current state of the art. *Rapid Prototyping Journal*, 16(2):130–137.

[Bidanda and Geng, 2016] Bidanda, B. and Geng, Z. (2016). Emerging trends in Reverse Engineering. In *2nd International Conference on Pro-AM Progress in Additive Manufacturing*.

[Biglino et al., 2016] Biglino, G., Capelli, C., Taylor, A. M., and Schievano, S. (2016). 3D Printing Cardiovascular Anatomy: A Single-Centre Experience. In Shishkovsky, I. V., editor, *New Trends in 3D Printing*, chapter 06. InTech, Rijeka.

[Birkfellner, 2015] Birkfellner, W. (2015). *Applied medical image processing: a basic course*. CRC Press.

[Borah et al., 2001] Borah, B., Gross, G. J., Dufresne, T. E., Smith, T. S., Cockman, M. D., Chmielewski, P. A., Lundy, M. W., Hartke, J. R., and Sod, E. W. (2001). Three-dimensional microimaging (MRI and CT), finite element modeling, and rapid prototyping provide unique insights into bone architecture in osteoporosis. *The Anatomical Record*, 265(2):101–110.

[Bortolotto et al., 2016] Bortolotto, C., Eshja, E., Peroni, C., Orlandi, M. A., Bizzotto, N., and Poggi, P. (2016). 3D Printing of CT Dataset: Validation of an Open Source and Consumer-Available Workflow. *Journal of Digital Imaging*, 29(1):14–21.

[Bracewell and Riddle, 1967] Bracewell, R. N. and Riddle, A. (1967). Inversion of fan-beam scans in radio astronomy. *The Astrophysical Journal*, 150:427.

[Brock, 2013] Brock, K. K. (2013). *Image processing in radiation therapy.* CRC Press.

[Camurri et al., 2014] Camurri, M., Vezzani, R., and Cucchiara, R. (2014). 3D Hough Transform for Sphere Recognition on Point Clouds. *Mach. Vision Appl.*, 25(7):1877–1891.

[Cao et al., 2006] Cao, M., Ye, C., Doessel, O., and Liu, C. (2006). Spherical parameter detection based on hierarchical Hough transform. *Pattern Recognition Letters*, 27(9):980 – 986.

[Chang and Chen, 2011] Chang, K.-H. and Chen, C. (2011). 3D shape engineering and design parameterization. *Computer-Aided Design and Applications*, 8:681–692.

[Chimento et al., 2011] Chimento, J., Highsmith, M. J., and Crane, N. (2011). 3D printed tooling for thermoforming of medical devices. *Rapid Prototyping Journal*, 17(5):387–392.

[Christoph and Neumann, 2012] Christoph, R. and Neumann, H. (2012). *X-ray Tomography in Industrial Metrology: Precise, Economical and Universal.* Verlag Moderne Industrie.

[Cignoni et al., 1998] Cignoni, P., Rocchini, C., and Scopigno, R. (1998). Metro: Measuring Error on Simplified Surfaces. *Computer Graphics Forum*, 17(2):167–174.

[Colombo et al., 2006] Colombo, G., Bertetti, M., Bonacini, D., and Magrassi, G. (2006). Reverse engineering and rapid prototyping techniques to innovate prosthesis socket design. In Corner, B. D., Li, P., and Tocheri, M., editors, *Three-Dimensional Image Capture and Applications*, volume 6056 of *SPIE Proceedings*, page 60560P. SPIE.

[Cormen et al., 2001] Cormen, T. H., Stein, C., Rivest, R. L., and Leiserson, C. E. (2001). *Introduction to Algorithms.* McGraw-Hill Higher Education, 2nd edition.

[Cramér and Wold, 1936] Cramér, H. and Wold, H. (1936). Some Theorems on Distribution Functions. *Journal of the London Mathematical Society*, s1-11:290–294.

[Dobbins and Godfrey, 2003] Dobbins, J. T. and Godfrey, D. J. (2003). Digital x-ray tomosynthesis: current state of the art and clinical potential. *Physics in Medicine and Biology*, 48(19):R65.

[Dössel and Buzug, 2014] Dössel, O. and Buzug, T. (2014). *Medizinische Bildgebung.* Biomedizinische Technik / [DGBMT, Deutsche Gesellschaft für Biomedizinische Technik im VDE]. Hrsg. von Ute Morgenstern und Marc Kraft. De Gruyter.

[Drosse et al., 2008] Drosse, I., Volkmer, E., Capanna, R., Biase, P. D., Mutschler, W., and Schieker, M. (2008). Tissue engineering for bone defect healing: An update on a multi-component approach. *Injury*, 39:S9 – S20.

[Drost and Ilic, 2015] Drost, B. and Ilic, S. (2015). Local hough transform for 3d primitive detection. In *3D Vision (3DV), 2015 International Conference on*, pages 398–406. IEEE.

[du Plessis et al., 2016] du Plessis, A., le Roux, S. G., and Guelpa, A. (2016). Comparison of medical and industrial X-ray computed tomography for non-destructive testing. *Case Studies in Nondestructive Testing and Evaluation*, 6, Part A:17 – 25.

[Elaraj, 2017] Elaraj, D. (2017). 3D Ultrasound. In *Advanced Thyroid and Parathyroid Ultrasound*, pages 59–66. Springer.

[Fastermann, 2014] Fastermann, P. (2014). *3D-Drucken: Wie die generative Fertigungstechnik funktioniert.* Technik im Fokus. Springer Vieweg, Berlin.

[Feldkamp et al., 1984] Feldkamp, L., Davis, L., and Kress, J. (1984). Practical cone-beam algorithm. *JOSA A*, 1(6):612–619.

[Fischler and Bolles, 1981] Fischler, M. A. and Bolles, R. C. (1981). Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Commun. ACM*, 24(6):381–395.

[Frey et al., 2012] Frey, M. A., Michaud, M., VanHouten, J. N., Insogna, K. L., Madri, J. A., and Barrett, S. E. (2012). Phosphorus-31 MRI of hard and soft solids using quadratic echo line-narrowing. *Proceedings of the National Academy of Sciences of the United States of America*, 109(14):5190–5195.

[Garland and Heckbert, 1998] Garland, M. and Heckbert, P. S. (1998). Simplifying surfaces with color and texture using quadric error metrics. In *Visualization'98. Proceedings*, pages 263–269. IEEE.

[Gebhardt, 2013] Gebhardt, A. (2013). *Generative Fertigungsverfahren: Additive Manufacturing und 3D Drucken für Prototyping - Tooling - Produktion.* Carl Hanser Verlag GmbH & Company KG.

[Gebhardt and Hötter, 2016] Gebhardt, A. and Hötter, J.-S., editors (2016). *Additive Manufacturing: 3D Printing for Prototyping and Manufacturing.* Carl Hanser Verlag GmbH & Company KG.

[Gelfand and Guibas, 2004] Gelfand, N. and Guibas, L. J. (2004). Shape Segmentation Using Local Slippage Analysis. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*, SGP '04, pages 214–223, New York, NY, USA. ACM.

[Gholizadeh, 2016] Gholizadeh, S. (2016). A review of non-destructive testing methods of composite materials. *Procedia Structural Integrity*, 1:50 – 57. XV Portuguese Conference on Fracture, PCF 2016, 10-12 February 2016, Paco de Arcos, Portugal.

[Gockel, 2006] Gockel, T. (2006). *Interaktive 3D-Modellerfassung mittels One-Shot-Musterprojektion und Schneller Registrierung.* Dissertation, Universität Karlsruhe, TH.

[Gomi, 2016] Gomi, T. (2016). X-ray Digital Tomosynthesis Imaging—Comparison of Reconstruction Algorithms in Terms of a Reduction in the Exposure Dose for Arthroplasty. *ARTHROPLASTY*, page 161.

[Grant, 1972] Grant, D. G. (1972). Tomosynthesis: A Three-Dimensional Radiographic Imaging Technique. *IEEE Transactions on Biomedical Engineering*, BME-19(1):20–28.

[Hagl, 2014] Hagl, R. (2014). *Das 3D-Druck-Kompendium: Leitfaden für Unternehmer, Berater und Innovationstreiber.* EBL-Schweitzer. Springer Fachmedien Wiesbaden.

[Harvey et al., 2014] Harvey, G., Gachagan, A., and Mutasa, T. (2014). Review of high-power ultrasound-industrial applications and measurement methods. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 61(3):481–495.

[Herment et al., 1987] Herment, A., Guglielmi, J., Dumee, P., Peronneau, P., and Delouche, P. (1987). Limitations of ultrasound imaging and image restoration. *Ultrasonics*, 25(5):267 – 273.

[Horn and Brooks, 1989] Horn, B. K. and Brooks, M. J. (1989). *Shape from shading.* MIT press.

[Hough, 1962] Hough, P. (1962). Method and means for recognizing complex patterns. *US Patent*, 3,069,654.

[Jansson, 2017] Jansson, A. (2017). *Only a Shadow: Industrial Computed Tomography Investigation, and Method Development, Concerning Complex Material Systems.* dissertation, Örebro University, School of Science and Technology.

[Jin et al., 2013] Jin, G. Q., Li, W. D., and Gao, L. (2013). An adaptive process planning approach of rapid prototyping and manufacturing. *Robotics and Computer Integrated Manufacturing*, 29:23–38.

[Jutras and De Zanche, 2016] Jutras, J.-D. and De Zanche, N. (2016). Sci-Thur PM - Colourful Interactions: Highlights 04: A Fast Quantitative MRI Acquisition and Processing Pipeline for Radiation Treatment Planning and Simulation. *Medical Physics*, 43(8Part3):4931–4932.

[Kaczmarz, 1937] Kaczmarz, S. (1937). Angenäherte Auflösung von Systemen linearer Gleichungen. *Bulletin International de l'Academie Polonaise des Sciences et des Lettres*, 35:355–357.

[Kak and Slaney, 2001] Kak, A. C. and Slaney, M. (2001). *Principles of computerized tomographic imaging.* SIAM.

[Kalender, 2011] Kalender, W. A. (2011). *Computed tomography: fundamentals, system technology, image quality, applications.* John Wiley & Sons.

[Kalender and Kyriakou, 2007] Kalender, W. A. and Kyriakou, Y. (2007). Flat-detector computed tomography (FD-CT). *European Radiology*, 17(11):2767–2779.

[Kasparova et al., 2013] Kasparova, M., Grafova, L., Dvorak, P., Dostalova, T., Prochazka, A., Eliasova, H., Prusa, J., and Kakawand, S. (2013). Possibility of reconstruction of dental plaster cast from 3D digital study models. *BioMedical Engineering OnLine*, 12(1):49.

[Katsevich et al., 2017] Katsevich, A., Rothermel, D., and Schuster, T. (2017). An improved exact inversion formula for solenoidal fields in cone beam vector tomography. *Inverse Problems*, 33(6):064001.

[Kazhdan et al., 2006] Kazhdan, M., Bolitho, M., and Hoppe, H. (2006). Poisson Surface Reconstruction. In *Proceedings of the Fourth Eurographics Symposium on Geometry Processing*, SGP '06, pages 61–70, Aire-la-Ville, Switzerland, Switzerland. Eurographics Association.

[Kouhi et al., 2008] Kouhi, E., Masood, S., and Morsi, Y. (2008). Design and fabrication of reconstructive mandibular models using fused deposition modeling. *Assembly Automation*, 28(3):246–254.

[Kulkarni et al., 2000] Kulkarni, P., Marsan, A., and Dutta, D. (2000). A review of process planning techniques in layered manufacturing. *Rapid Prototyping Journal*, 6(1):18–35.

[Kumar et al., 2013] Kumar, A., Jain, P., and Pathak, P., editors (2013). *Reverse Engineering in Product Manufacturing: an Overview.* DAAAM International Scientific Book 2013.

[Lin et al., 2007] Lin, F., Seah, H., Wu, Z., and Ma, D. (2007). Voxelization and fabrication of freeform models. *Virtual and Physical Prototyping*, 2(2):65–73.

[Liu et al., 2003] Liu, G., Wong, Y., Zhang, Y., and Loh, H. (2003). Error-based segmentation of cloud data for direct rapid prototyping. *Computer-Aided Design*, 35(7):633 – 645.

[Lorensen and Cline, 1987] Lorensen, W. E. and Cline, H. E. (1987). Marching cubes: A high resolution 3D surface construction algorithm. In *ACM siggraph computer graphics*, volume 21, pages 163–169. ACM.

[Louis, 2016] Louis, A. K. (2016). Exact cone beam reconstruction formulae for functions and their gradients for spherical and flat detectors. *Inverse Problems*, 32(11):115005.

[Lueth, 2017] Lueth, T. C. (2017). Matlab-Toolbox for Automated Design of Solid Geometries (CAD) and Multi-Body-Simulation. https://www.mimed.mw.tum.de/research/tim-lueths-research/.

[Muja and Lowe, 2009] Muja, M. and Lowe, D. G. (2009). Fast approximate nearest neighbors with automatic algorithm configuration. In *In VISAPP International Conference on Computer Vision Theory and Applications*, pages 331–340.

[Müller et al., 2013] Müller, P., Cantatore, A., Andreasen, J. L., Hiller, J., and Chiffre, L. D. (2013). Computed Tomography as a Tool for Tolerance Verification of Industrial Parts. *Procedia CIRP*, 10:125 – 132.

[Nevin and Mehrdad, 2014] Nevin, H. and Mehrdad, H. (2014). Deposition direction-dependent failure criteria for fused deposition modeling polycarbonate. *Rapid Prototyping Journal*, 20:221–227.

[Pertuz et al., 2013] Pertuz, S., Puig, D., and Garcia, M. A. (2013). Analysis of Focus Measure Operators for Shape-from-focus. *Pattern Recogn.*, 46(5):1415–1432.

[Pradhan et al., 2010] Pradhan, R., Kumar, S., Agarwal, R., Pradhan, M. P., and Ghose, M. (2010). Contour line tracing algorithm for digital topographic maps. *International Journal of Image Processing (IJIP)*, 4:156–163.

[Prša et al., 2014a] Prša, J., Irlinger, F., and Lueth, T. C. (2014a). Algorithm for Detecting and Solving the Problem of Under-filled Pointed Ends Based on 3D Printing Plastic Droplet Generation. In *ASME International Mechanical Engineering Congress and Exposition*, Montreal, Canada.

[Prša et al., 2015a] Prša, J., Jain, R., Irlinger, F., and Lueth, T. C. (2015a). Automatic, reference-free and conformity-oriented evaluation and interpretation of CT-models. In *2015 International Conference on Computing and Network Communications (CoCoNet)*, pages 524–529.

[Prša et al., 2014b] Prša, J., Mueller, J., Irlinger, F., and Lueth, T. C. (2014b). Evaluation of the Infill Algorithm for Trajectory Planning of Pointed Ends for Droplet-Generating 3D Printers. In *IEEE International Conference on Robotics and Biomimetics, publisher = December*, Bali, Indonesia.

[Prša et al., 2013] Prša, J., Schwaiger, J., Irlinger, F., and Lueth, T. C. (2013). Dense 3D-Packing Algorithm for Filling the Offset Contours of a New Printing Process Based on 3D Plastic Droplet Generation. In Shenzhen, C., editor, *IEEE International Conference on Robotics and Biomimetics (ROBIO).*

[Prša et al., 2015b] Prša, J., Sobreviela, J., Irlinger, F., and Lueth, T. C. (2015b). Software tool for detection and filling of voids as a part of tool-path strategy development for droplet generating 3D printers. In *2015 International Conference on Computer, Information and Telecommunication Systems (CITS).*

[Qiu et al., 2001] Qiu, D., Langrana, N. A., Danforth, S. C., Safari, A., and Jafari, M. (2001). Intelligent Toolpath for Extrusion-based LM Process. *Rapid Prototyping Journal*, 7(1):18–23.

[Qiu and Seeram, 2016] Qiu, D. and Seeram, E. (2016). Does iterative reconstruction improve image qual-ity and reduce dose in computed tomography. *Radiology - Open Journal*, 1(2):42–54.

[Raja and Fernandes, 2007] Raja, V. and Fernandes, K. (2007). *Reverse Engineering: An Industrial Perspective.* Springer Series in Advanced Manufacturing. Springer London.

[Reidl et al., 2016] Reidl, M., Ockert, A., and Daniela, G. (2016). Fachgespräch Carl Zeiss 3D Metrology. Technical report, Technical University of Munich.

[Reinhart et al., 2004] Reinhart, C., Poliwoda, C., Guenther, T., Roemer, W., Maass, S., and Gosch;, C. (2004). Modern voxel based data and geometry analysis software tools for industrial CT. In *16th World Conference on Nondestructive Testing* .

[Rinck, 2017] Rinck, P. (2017). *Magnetic Resonance in Medicine.* The Basic Textbook of the European Magnetic Resonance Forum, 10th edition. E-version 10.1 beta.

[Rizzini et al., 2014] Rizzini, D. L., Oleari, F., Atti, A., Aleotti, J., and Caselli, S. (2014). Unsupervised Range Image Segmentation and Object Recognition Using Feature Proximity and Markov Random Field. In *Intelligent Autonomous Systems*, pages 807–820.

[Ronfard and Rossignac, 1996] Ronfard, R. and Rossignac, J. (1996). Full-range approximation of triangulated polyhedra. In *Computer Graphics Forum*, volume 15(3), pages 67–76. Wiley Online Library.

[Rossignac and Borrel, 1993] Rossignac, J. and Borrel, P. (1993). Multi-resolution 3D approximations for rendering complex scenes. *Modeling in Computer Graphics: methods and applications*, 465.

[Schnabel et al., 2007] Schnabel, R., Wahl, R., and Klein, R. (2007). Efficient RANSAC for Point-Cloud Shape Detection. *Computer Graphics Forum*, 26(2):214–226.

[Schroeder et al., 1992] Schroeder, W. J., Zarge, J. A., and Lorensen, W. E. (1992). Decimation of triangle meshes. In *ACM Siggraph Computer Graphics*, volume 26, pages 65–70. ACM.

[Schwaiger et al., 2013] Schwaiger, J., Lueth, T. C., and Irlinger, F. (2013). G-Code Generation for a New Printing Process Based on 3D Plastic Polymer Droplet Generation. *ASME International Mechanical Engineering Congress and Exposition*, pages 15–21.

[Shaw, 2014] Shaw, C. (2014). *Cone Beam Computed Tomography.* Imaging in medical diagnosis and therapy. CRC Press.

[Shontz and Nistor, 2013] Shontz, S. M. and Nistor, D. M. (2013). *CPU-GPU Algorithms for Triangular Surface Mesh Simplification*, pages 475–492. Springer Berlin Heidelberg, Berlin, Heidelberg.

[Skala et al., 2013] Skala, V., Pan, R., and Nedved, O. (2013). Simple 3D Surface Reconstruction Using Flatbed Scanner and 3D Print. In *SIGGRAPH Asia 2013 Posters*, SA '13, pages 7:1–7:1, New York, NY, USA. ACM.

[Sun et al., 2012] Sun, W., Brown, S., and Leach, R. (2012). An overview of industrial X-ray computed tomography. NPL Report ENG 32, Nuffield College, Oxford, UK, Discussion paper.

[Talton, 2004] Talton, J. O. (2004). A short survey of mesh simplification algorithms. Technical report, University of Illinois at Urbana-Champaign.

[Tanabe, 1971] Tanabe, K. (1971). Projection method for solving a singular system of linear equations and its applications. *Numerische Mathematik*, 17(3):203–214.

[Tarabanis, 2001] Tarabanis, K. A. (2001). Path planning in the Proteus rapid prototyping system. *Rapid Prototyping Journal*, 7(5):241–252.

[Tedia and Williams, 2016] Tedia, S. and Williams, C. B. (2016). Manufacturability analysis tool for additive manufacturing using voxel-based geometric modeling. *Solid Freeform Fabrication 2016: Proceedings of the 27th Annual International Conference.*

[Tiller and Hanson, 1984] Tiller, W. and Hanson, E. G. (1984). Offsets of Two-Dimensional Profiles. *IEEE Computer Graphics and Applications*, pages 36–46.

[Tita, 2007] Tita, R. (2007). *Variable isozentrische Steuerung für einen Standard-C-Bogen mit echtzeitfähiger 3D-Rekonstruktion.* Dissertation, Technische Universität München.

[Turk, 1992] Turk, G. (1992). Re-tiling polygonal surfaces. *ACM SIGGRAPH Computer Graphics*, 26(2):55–64.

[Tuy, 1983] Tuy, H. K. (1983). An Inversion Formula for Cone-Beam Reconstruction. *SIAM Journal on Applied Mathematics*, 43(3):546–552.

[van der Glas et al., 2002] van der Glas, M., Vos, F. M., Botha, C. P., and Vossepoel, A. M. (2002). Determination of position and radius of ball joints. In *Proceedings of the SPIE international symposium on medical imaging*, volume 4684, pages 1571–1577.

[Varslot et al., 2011] Varslot, T., Kingston, A., Myers, G., and Sheppard, A. (2011). High-resolution helical cone-beam micro-CT with theoretically-exact reconstruction from experimental data. *Medical Physics*, 38(10):5459–5476.

[Voicu et al., 2013] Voicu, A. C., Gheorghe, I. G., Badita, L. L., and Cirstoiu, A. (2013). 3D Measuring of Complex Automotive Parts by Multiple Laser Scanning. In *Innovative Manufacturing Engineering*, volume 371 of *Applied Mechanics and Materials*, pages 519–523. Trans Tech Publications.

[Von Ramm and Smith, 1987] Von Ramm, O. T. and Smith, S. W. (1987). Three-dimensional imaging system. *US Patent*, 4694434.

[Wang, 2010] Wang, W. (2010). *Reverse Engineering: Technology of Reinvention.* CRC Press.

[Willis et al., 2007] Willis, A., Speicher, J., and Cooper, D. B. (2007). Rapid prototyping 3D objects from scanned measurement data. *Image and Vision Computing*, 25(7):1174 – 1184. Computer Vision Applications.

[Wohlers et al., 2016] Wohlers, T., Associates, W., Campbell, R., and Caffrey, T. (2016). *Wohlers Report 2016: 3D Printing and Additive Manufacturing State of the Industry : Annual Worldwide Progress Report.* Wohlers Associates.

[Wu et al., 2004] Wu, Y., Wong, Y., Loh, H., and Zhang, Y. (2004). Modelling cloud data using an adaptive slicing approach. *Computer-Aided Design*, 36(3):231 – 240.

[Xiong et al., 2008] Xiong, Y.-Y., Jiao, T., and Zhang, F.-Q. (2008). Application of three-dimensional structured-light measurement technique and rapid prototyping technique in maxillofacial prosthetics. *Journal of Clinical Rehabilitative Tissue Engineering Research*, 12(9):1705–1708.

[Xu et al., 2017] Xu, Y., Tuttas, S., Hoegner, L., and Stilla, U. (2017). Geometric Primitive Extraction From Point Clouds of Construction Sites Using VGS. *IEEE Geoscience and Remote Sensing Letters*, 14(3):424–428.

[Yi et al., 2014] Yi, B., Liu, Z., Tan, J., Cheng, F., Duan, G., and Liu, L. (2014). Shape recognition of CAD models via iterative slippage analysis. *Computer-Aided Design*, 55:13 – 25.

[Yu and Leng, 2016] Yu, L. and Leng, S. (2016). *Image reconstruction Techniques.* Radiation Safety in Adult Medical Imagin, Image Wiselys.

[Zhang et al., 2015] Zhang, Y., Gibson, G. M., Hay, R., Bowman, R. W., Padgett, M. J., and Edgar, M. P. (2015). A fast 3D reconstruction system with a low-cost camera accessory. *Scientific Reports*, 5.