

INSTITUTE OF APPLIED MECHANICS (CE)

CHAIR OF CONTINUUM MECHANICS

SIMULATION TECHNOLOGY DEGREE COURSE

Master thesis

Submitted to the University of Stuttgart

Reconstruction of μ XRCT data sets using the ASTRA toolbox

First Examiner

Prof. Dr.-Ing. Holger STEEB

Institute of Applied Mechanics (CE)

Second Examiner

apl. Prof. Dr.-Ing. Holger CLASS

Institute for Modelling Hydraulic and
Environmental Systems

Supervisor

M. Sc. Matthias RUF

Institute of Applied Mechanics (CE)

Submitted by

Author

Paul VOLAND

Matriculation number

2872964

SimTech number

65

E-Mail

st107953@stud.uni-stuttgart.de

Submission date

19 June 2020

Declaration of originality

Hereby, I confirm that I have independently written the present work, that I have not used sources other than those specified and have marked all statements taken verbatim or analogously from other works, that the submitted work has neither completely nor in substantial part been subject to any other examination procedure, that I have not published the work in full or in part and that the electronic copy agrees with the other copies.

Stuttgart, 19 June 2020

signature of the author (Paul Voland)

Preamble

The present master thesis “Reconstruction of μ XRCT data sets using the ASTRA toolbox” serves to obtain the master’s degree in the degree course M. Sc. Simulation Technology at the University of Stuttgart. Therefore, all sections shall be evaluated in accordance with the applying examination regulations of the year 2016¹.

¹<https://www.student.uni-stuttgart.de/pruefungsorganisation/pruefungsordnung/master-of-science/#S>

Abstract

X-ray computed tomography (XRCT) and especially micro X-ray computed tomography (μ XRCT) represent key tools to get an insight into the internal structure of an object under investigation. While XRCT enjoys great popularity in its application for medical imaging, μ XRCT is also of great interest for other research fields like materials science due to its higher spatial resolution. The objective of this master thesis is to set up an algorithmic workaround to be able to reconstruct real-world μ XRCT data sets recorded in the open, modular and flexible XRCT system of the Institute of Applied Mechanics (CE) at the University of Stuttgart. Therein, the ASTRA toolbox is used as a basis, providing main routines for reconstruction tasks such as the reconstruction algorithms. This work prepares the indispensable theoretical underpinning concerning X-ray physics, common experimental scanning setups and takes a detailed look at the mathematics of the reconstruction algorithms available inside the ASTRA toolbox. Coming to the concrete implementation, image processing and filtering techniques play a major role to achieve meaningful reconstruction results. Several studies dealing with qualitative aspects which are of central importance for the reconstruction of real-world μ XRCT data sets conclude the present master thesis. It also discusses under which restrictions it is possible to reconstruct full 3D data sets obtained with cone or helical cone beam scanning.

Contents

List of Figures	3
List of Tables	5
List of Abbreviations	7
List of frequent Symbols	9
1. Introduction	11
2. Physics on X-rays	13
2.1. Generation of X-rays	13
2.2. Interaction mechanisms with matter	15
2.3. Detection of X-rays	16
3. Selected experimental scan setups	19
3.1. Parallel beam scanning	19
3.2. Fan beam scanning	20
3.3. Cone beam scanning	21
3.4. Helical cone beam scanning	21
3.5. Remarks on projection acquisition and coordinate systems	22
4. Reconstruction fundamentals	25
4.1. Radon transformation and Fourier slice theorem	25
4.1.1. Radon transformation	25
4.1.2. Fourier slice theorem	27
4.1.3. Remarks and challenges	29
4.2. 2D reconstruction methods for parallel beam scanning	30
4.2.1. Simple backprojection	31
4.2.2. Filtered backprojection	32
4.2.3. Aspects of discretization and implementation	34
4.3. 2D reconstruction methods for fan beam scanning	36
4.4. 3D reconstruction methods for cone beam scanning	41
4.5. Algebraic reconstruction methods in 2D and 3D	48
4.5.1. Simultaneous iterative reconstruction technique (SIRT)	49
4.5.2. Conjugate gradients for least squares (CGLS)	50
5. The ASTRA toolbox	55
5.1. Main components of the ASTRA toolbox	55
5.2. Application to the experimental setups	58
6. Image processing and filtering techniques	61
6.1. Normalization	61
6.2. Sinograms	63
6.3. Cropping and region of interest (ROI) mask	64

6.4. Correction of center of rotation (COR) misalignments	65
6.4.1. Characterization of possible center of rotation (COR) misalignments	66
6.4.2. Correction of misalignments through center of rotation (COR) shift	68
6.4.3. Correction of misalignments through center of rotation (COR) tilt	68
6.5. Beam hardening correction (BHC)	69
7. Results, comparison and discussion	75
7.1. Processing of a full cone beam scanning data set	75
7.1.1. Introduction and utilizability of the overlap	76
7.1.2. Remarks on occurred differences between 2D and 3D reconstruction	81
7.1.3. Application to an irregular real-world data set	83
7.2. Treatment of helical cone beam scanning data	87
8. Conclusion	93
A. Appendix	95
Bibliography	109

List of Figures

2.1. illustration of the three main mechanisms to generate X-ray radiation	13
2.2. X-ray spectra for a X-ray tube voltage of 100 kV	14
2.3. abstracted functionality of a scintillation X-ray detector	17
3.1. simplified representation of a 2D or 3D parallel beam scanning scenario	19
3.2. simplified representation of a 2D fan beam scanning scenario	20
3.3. simplified representation of a 3D cone beam scanning scenario	21
3.4. simplified representation of a 3D helical cone beam scanning scenario	22
3.5. comparison of theoretical and empirical acquisition of projection data	23
4.1. introduction of the rotated polar ($\mathbf{e}_\xi, \mathbf{e}_\eta$) coordinate system	26
4.2. interactive structure to clarify the coherence of the various spaces used in the discussion of the Fourier slice theorem	28
4.3. flow chart to emphasize the overall benefit of the Fourier slice theorem as a linking element in computed tomography	29
4.4. continuous and discrete spectral projection data in two competitive coordinate systems	30
4.5. blurring introduced by the usage of the ASTRA simple backprojection algorithm BP .	32
4.6. geometrical situation for rebinning of a curved detector array	37
4.7. geometrical situation for rebinning of a flat detector array	38
4.8. global overview of the geometry necessary for the derivation of the FDK reconstruction algorithm for cone beam scanning	42
4.9. illustration of the orientation of the new fan-specific coordinate system ($\mathbf{e}_\sigma, \mathbf{e}_\eta, \mathbf{e}_\xi$) by its spanning normal vectors $\{\mathbf{n}_\sigma, \mathbf{n}_\eta, \mathbf{n}_\xi\}$	43
4.10. focussing several right-angled triangles from Figure 4.8	44
4.11. additional illustration of geometrical relations used to derive the FDK algorithm	47
5.1. schematic overview of the ASTRA toolbox design	56
5.2. auxiliary illustration to obtain a deeper understanding of the vectorization	59
6.1. normalization steps for one projection image of a stepped brass cylinder phantom with $M = 16.50$	62
6.2. exemplary sinograms generated from corresponding normalized projection images . .	63
6.3. reconstructions of an asphalt sample with $M = 2.99$ for different COR parameters . .	65
6.4. overview of possible center of rotation (COR) misalignments	67
6.5. comparison of reconstructions for the small diameter region of a stepped brass cylinder phantom ($M = 33.02$) with regard to the application of BHC	70
6.6. numerical studies of the gray value trend for each coefficient in $\{d, e, f, g\}$ independently	72
6.7. comparison as in Figure 6.5 for the large diameter region	73
7.1. auxiliary sketch of the geometrical situation for the definition of <code>nec_slice_count</code> . .	76
7.2. symbolic explanation of the algorithmic treatment of overlap regions	77
7.3. numerical study of the gray value trends along the slice's stack height for varying geometric magnifications	79
7.4. correlation between geometric magnification and required overlap divisor for a proper reconstruction	81

7.5. numerical study of the signal-to-noise ratio SNR for the stepped brass cylinder phantom with $M = 16.5$	82
7.6. numerical study of the mean gray value trends for selected slices of reconstructions for the open-pored asphalt concrete sample with $M = 2.99$	84
7.7. reconstructed images of the open-pored asphalt concrete sample with $\text{overlap_div} = 2$	85
7.8. sectional views of the open-pored asphalt concrete sample reconstructed with $\text{overlap_div} = 2$ along the \mathbf{e}_z -direction	86
7.9. schematic, exemplary representation of all imaginable combinations that affect the orientation of the ASTRA projection geometry for helical cone beam scanning	88
7.10. visualization of the underlying exact voxel data in different heights	89
7.11. sinograms and reconstructions of the test phantom generated with the specified helical cone beam scanning projection geometry	90
A.1. numerical studies of the gray value trends along the slice's stack height for $M = 4.95$	96
A.2. numerical studies of the gray value trends along the slice's stack height for $M = 16.5$	100
A.3. numerical studies of the gray value trends along the slice's stack height for $M = 33.02$	104

List of Tables

7.1. line-up of slice numbers corresponding to increasing overlap divisors for fixed num_
slices_per_part = 60 80

List of Abbreviations

ART	algebraic reconstruction technique
BHC	beam hardening correction
BP	(simple) backprojection
CGLS	conjugate gradients for least squares
COR	center of rotation
CPU	central processing unit
FBP	filtered backprojection
FDK	Feldkamp, Davis and Kress
FFT	fast Fourier transform
FOV	field of view
FST	Fourier slice theorem
GPU	graphics processing unit
GUI	graphical user interface
ID	identifier
MRI	magnetic resonance imaging
μ XRCT	micro X-ray computed tomography
PSF	point-spread function
RAM	random-access memory
ROI	region of interest
SART	simultaneous algebraic reconstruction technique
SDLS	steepest decent for least squares
SIRT	simultaneous iterative reconstruction technique
SNR	signal-to-noise ratio
TIFF	tagged image file format
USCT	ultrasound computed tomography
XRCT	X-ray computed tomography

XRT X-ray tomography

List of frequent Symbols

\mathbf{d}	vector from the global origin \mathcal{O} to the middle pixel of the detector
d_{od}	distance from the global origin \mathcal{O} to the detector's center
d_{sd}	distance from the X-ray source to the detector's center
d_{so}	distance from the X-ray source to the global origin \mathcal{O}
$E_{(p)}$	(photon) energy
\mathbf{e}_\bullet	unit vector of a specific coordinate system with coefficients \bullet
I	X-ray intensity
M	geometric magnification
μ	attenuation coefficient
N_x/N_z	number of (detector) pixels in $\mathbf{e}_x/\mathbf{e}_z$ -direction
N_θ	number of projection angles θ
\mathbf{n}_\bullet	normal vector pointing in \mathbf{e}_\bullet -direction
\mathcal{O}	global origin of each coordinate system located in the object's center
p	projection (gray) value
p_x/p_z	detector pixel size in $\mathbf{e}_x/\mathbf{e}_z$ -direction
\tilde{p}_x/\tilde{p}_z (v_x/v_z)	voxel size in $\mathbf{e}_x/\mathbf{e}_z$ -direction
\mathbf{s}	vector from the global origin \mathcal{O} to the X-ray source
θ	projection angle
\mathbf{u}	vector from the middle pixel of the detector to the next pixel on the right
\mathbf{v}	vector from the middle pixel of the detector to the next pixel at the top

1. Introduction

We understand tomography by its definition as an imaging technique using sectionings of an object to draw conclusions about the internal structures from these slices. The most popular representatives of tomographic procedures are X-ray computed tomography (XRCT), magnetic resonance imaging (MRI) and ultrasound computed tomography (USCT). Each of them produces sectional images, widely known as projections, in which the interior of the object cannot be identified free from superposition effects. The tomographic method applied in this master thesis will be X-ray computed tomography, for which we firstly need a clear definition of terms: As the title already suggests, the data sets we make use of are obtained by micro X-ray computed tomography (μ XRCT). The difference between outdated X-ray tomography (XRT) and XRCT is that in XRT the projections (better known as radiographs in this case) had to be recorded analogously using a X-ray film, whereas nowadays digital detection of the radiation in XRCT is a standard procedure. In particular, the word component “computed” in XRCT refers to the fact that the orientation of the scanning installation, the generation of a number of radiographs and all the calculations in conjunction with the key reconstruction task for the sample are performed virtually using computer technology. Of course XRCT uses Röntgen radiation (and not other types of excitations like magnetic fields in MRI or ultrasound waves in USCT), and finally the prefix “ μ ” indicates that the resolution of the resulting tomographic reconstructions will be in a micrometer range.

In XRCT, the collected radiographs (more likely named projections in this context) are measured from different projection angles. Depending on the area of application, either the sample or the X-ray source and detector rotate to achieve variable projection images. The great advantages of XRCT lies in the fact that the insight into the object of interest can be gained in a noninvasive and nondestructive way. The basic prerequisite for XRCT is that the method is very sensitive to small changes in the X-ray absorption inside the object, leading to a reliable distinction between eventual internal structures. As presented in (Stock, 2008), the main utilization of XRCT following from these benefits is clearly in human (or veterinary) medical imaging for diagnostic and therapeutic purposes. The strength of XRCT for medical use is the speed (caused by the low exposure times) and the ability to demarcate tissues that only differ very slightly in its physical density (see Kasban, El-Bendary, and Salama, 2015). The second big use case of XRCT is industrial computed tomography, where internal inspection of the produced components stands in the foreground (cf. Sun, Brown, and Leach, 2012). It is necessary to impose higher radiation doses to check for production defects or to analyze assemblies within the inorganic materials. Of course, XRCT can also serve research in the fields of materials science (for one example, consider Bale et al., 2011) as well as biological or physical science, or can even be used in geological (Cnudde and Boone, 2013) or archaeological challenges. In general, it can be said that XRCT is a powerful tool to make the inside of objects accessible.

First of all, the original objective of this master thesis was to deal with the ASTRA toolbox as an alternative and relatively new open source software solution for reconstruction tasks. Alternative herein refers to the fact that the Institute of Applied Mechanics (CE) at the University of Stuttgart used to reconstruct projection data with the commercial software Octopus Reconstruction, whose sale was stopped in 2019 - so the need for another reconstruction tool is quite obvious. But also the additional features of the ASTRA toolbox like a variety of applicable reconstruction algorithms and the geometric flexibility (see chapter 5) shall be tested within this elaboration. To be able to

carry this out, a huge amount of preparational understanding has to be acquired by the user. First of all, chapter 2 demonstrates the physics on X-rays underlying any generated radiograph of an object under investigation. Chapter 3 depicts selected experimental scan setups for XRCT only and explains how exactly projection data sets are gathered in practice. The reconstruction fundamentals in chapter 4 play a very central role not only for this thesis, but also for the success of XRCT in general, since reconstruction is the crucial point on the way to meaningful resulting images for the internal structures of the sample. These cover the absolutely necessary previous knowledge on theoretical aspects of analytical (sections 4.1 to 4.4) and algebraic (section 4.5) μ XRCT reconstruction methods in 2D and 3D without claim of completeness. Since real-world projection data sets, as created in the open, modular and flexible XRCT system of the Institute of Applied Mechanics (CE) at the University of Stuttgart (for a detailed description of the in-house XRCT lab, see Ruf and Steeb, 2020a), raise additional challenges which are not included from the start in the toolbox routines, chapter 6 developed incidentally, offering some input concerning image processing and filtering. The sections on the correction of center of rotation (COR) misalignments in 6.4 and on beam hardening correction (BHC) in 6.5 are of particular interest here. To what extent the ASTRA toolbox can reconstruct more than just theoretically motivated projection data sets and what limitations and particularities appear in the process will be clarified in chapters 5 to 7, which involve and discuss some studies carried out with the ASTRA toolbox. Section 7.1 deals with the challenges on the way to processing of a full cone beam scanning data set and analyzes the final results for an irregular real-world data set, while section 7.2 elucidates the treatment of helical cone beam scanning data.

2. Physics on X-rays

This second chapter should give a brief overview over the underlying physical principles that make the insight into objects possible at all, using the μ XRCT scanning technique. The relevant phenomena for this can be separated into the generation of X-rays in section 2.1, the interaction mechanisms with matter (for example the positioned material sample) in section 2.2 and finally the detection of the remaining X-ray radiation in section 2.3. The book (Carmignato, Dewulf, and Leach, 2018) notably served as an inspiration for this chapter.

2.1. Generation of X-rays

X-rays (in the European usage mostly called Röntgen rays after their discoverer Wilhelm Röntgen) are electromagnetic photon waves. In general, the energy of a single photon E_p , also named photon energy, is proportional to its stimulation frequency ν as given by the relation

$$E_p = h\nu = h\frac{c}{\lambda}. \quad (2.1)$$

The constant h in the familiar Planck-Einstein relation (2.1) is the so called Planck constant, $h = 6.63 \cdot 10^{-34}$ Js. In the reformulation with the wave's wavelength λ on the right side, the constant c is the speed of light in vacuum, $c = 3 \cdot 10^8$ m/s. The characteristic wavelengths of X-rays range from 0.01 to 10 nm, and the radiation is called hard respectively soft radiation considering the energetic level attached to these wavelengths from equation (2.1).

To describe the generation of X-rays, we restrict ourselves to a classical polychromatic X-ray tube like the one used in our in-house XRCT lab to provide projections from a cone beam scanning setup. The generation of Röntgen radiation is mainly a two step mechanism: in the first step, the thermionic effect is exploited to generate an electron beam at the cathode of the X-ray tube, which is aimed at the target anode, where the electron beam interacts with the anode material in the second step and partially emits Röntgen radiation.

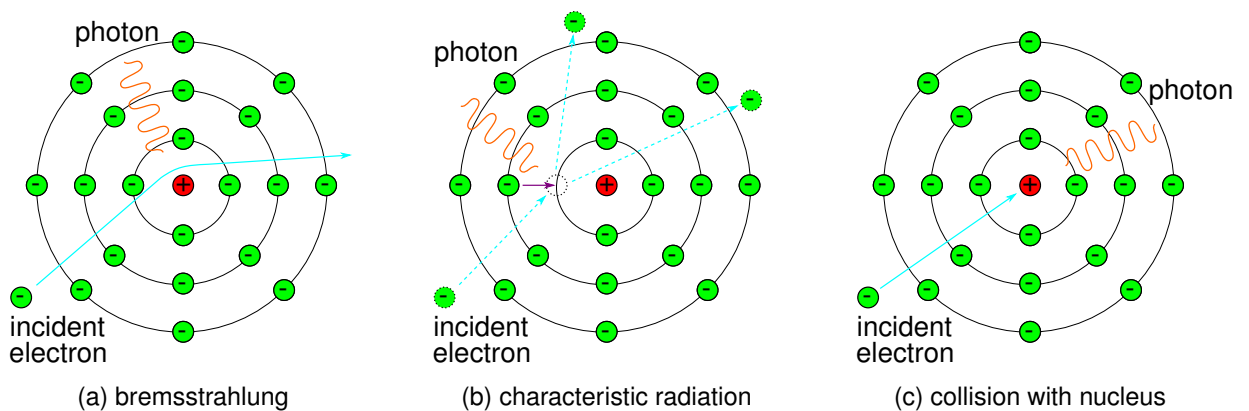


Figure 2.1.: illustration of the three main mechanisms to generate X-ray radiation

A little more detailed view on the X-ray tube shows that both the cathode and the anode are evacuated in a high vacuum chamber. The cathode is made up of a tungsten filament (wolfram) which is heated using the Joule effect at low voltage. As a result of the increasing temperature, the filament radiates electrons who have overcome the binding energy to their atoms by higher kinetic energy. This is called the thermionic effect, and the evolving electron beam is focussed on the target afterwards. The anode itself comprises a metal part, often again tungsten or molybdenum, and a thicker part around the main target to facilitate heat dissipation, mostly made up of copper.

When the electron beam hits the target material, 99 % of the present energy is transformed to heat originating from ionisation of target atoms. Only 1 % of the energy produces Röntgen radiation, and this is obtained by three different mechanisms: The main mechanism can be indicated with the german word *bremssstrahlung* and describes the deceleration of fast incident electrons in target atoms, see Figure 2.1(a). Also of interest is the second mechanism (compare Figure 2.1(b)), which can only be observed in small regions of photon energy. X-rays are emitted when a vacant electron place in an inner shell of a target atom is refilled with an outer shell electron, inducing energy loss to photons. The last and very rare mechanism depicted in Figure 2.1(c) is that an incident electron directly collides with a target atoms nucleus, emitting photons at a very high energy level.

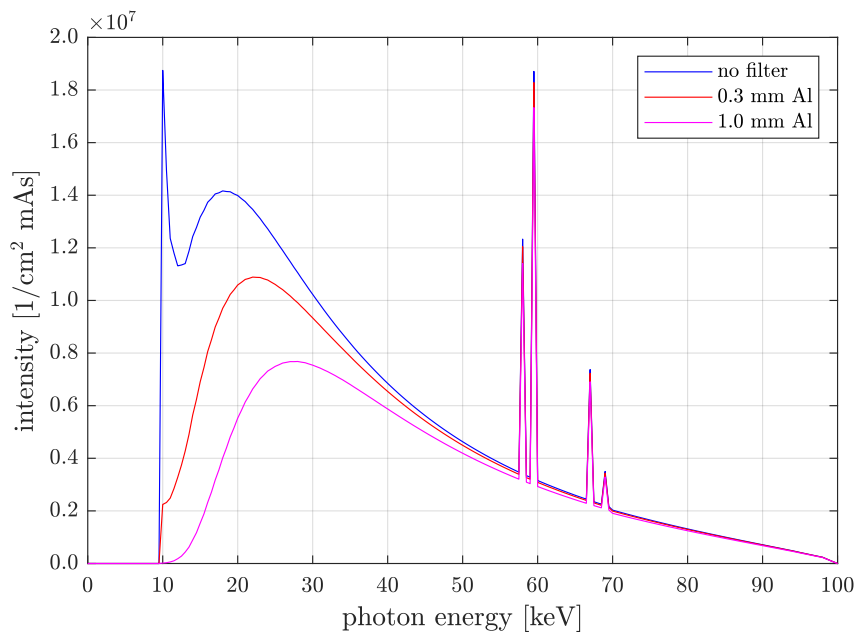


Figure 2.2.: X-ray spectra for a X-ray tube voltage of 100 kV, created with the software SpekCalc (Poludniowski, Landry, DeBlois, Evans, and Verhaegen, 2009; the start of the trends only at 10 keV is software-inherent)

We call a Röntgen radiation monochromatic if all enclosed photons own identical energy levels. On the contrary, a polychromatic X-ray tube produces a bulk of photons with basically arbitrary energy levels, and the denomination of the X-ray radiation as a spectrum on the lines of visible light is traceable since differing photon energies are affiliated with varying wave lengths according to equation (2.1). To support the comprehension of the three generating physical principles, Figure 2.2 illustrates polychromatic X-ray spectra for three different situations: Each curve predicts the intensity (which can be understood as a statistical frequency) of the Röntgen radiation with increasing photon energy levels, in which the applied X-ray tube voltage is 100 kV. The only difference in between them is that the red and magenta curves correspond to a physical filtering procedure with aluminum disks of dissimilar thickness, while the blue spectrum remains unfiltered. The continuous share

of each trend indicates the bremsstrahlung (cf. Figure 2.1(a)) because this phenomenon arises independent of the photon energy in principle. The peaks in the region of 55 to 70 keV photon energy in Figure 2.2 symbolize the characteristic radiation (cf. Figure 2.1(b)), and the location of the peaks is connected to the electron shell where the succeeding electron comes from. Collisions with a target atom's nucleus like in Figure 2.1(c) cannot be recognized in the graph because of its rareness - the vanishing intensity values for the highest photon energies around 100 keV confirm this statement.

2.2. Interaction mechanisms with matter

After a X-ray beam has been generated in the X-ray tube, the X-rays are exposed to the object sample of interest in a way that the intensity of the beam reduces exponentially with increasing object depth. The so called attenuation includes changes of the number, energy and direction of the photons comprising the beam. There are some reasons for this attenuation, of which we only want to take a short look at two thereof.

The first and also central cause for X-ray attenuation in matter is the photoelectric effect, which takes place at lower photon energy levels. Here, the incident photon with energy higher than the binding energy of an electron in the sample object effects the ejection of a lower shell electron, now named photoelectron. We can find the energy balance

$$h\nu = E_p = E_b + E_{pe}, \quad (2.2)$$

where E_b is the surmounted electron binding energy and E_{pe} the energy of the freed photoelectron. On this occasion, the photon is absorbed and the matter's affected atom stays ionised until the vacant place in the lower shell is filled again. A side effect when refilling the hole of the ionised atom is that a fluorescence photon is emitted in a similar way as presented in Figure 2.1 (b). The whole attenuation caused by the photoelectric effect depends on both the material and beam properties: How strongly the X-ray beam intensity is weakened is proportional to the object's atomic number Z and the photon energy E_p of the rays. There exists the thumb rule

$$\mu_{pe} \propto Z^4 \lambda^3 \quad (2.3)$$

with the attenuation coefficient μ_{pe} of the photoelectric effect and the wavelength λ belonging to a certain photon energy level.

The second noteworthy reason for the X-ray attenuation in matter is called Compton Scattering. Herein, a photon of middle energy level (but of course $E_p \gg E_b$) interacts with a bound electron of the material. The photon partially transfers its energy to the bound electron so that the electron is kicked out of the atom, whereas the photon continues with lower energy, shifted wavelength and new direction compared to before. The energy balance

$$h\nu = E_p = E_e + E_{p'} \quad (2.4)$$

holds true with the new energy of the deflected photon $E_{p'}$ and the energy of the ejected, so called Compton electron E_e .

After putting both effects together, the resulting attenuation coefficient is given by

$$\mu = \mu_{pe} + \mu_{compt}.$$

With this relation, we can continue to state the central law of Beer-Lambert, which describes the progress of attenuation of X-rays passing through matter. Let us first assume to have a monochromatic incident beam and a homogeneous material, so that the ordinary differential equation

$$\frac{dI}{I(x)} = -\mu dx \quad (2.5)$$

gives the spatial evolution of the intensity I of the X-ray in one dimension. Since the attenuation coefficient μ is constant for a homogeneous material, equation (2.5) integrates to

$$I(x) = I_0 e^{-\mu x} \quad (2.6)$$

with the passed distance through the object x and the initial X-ray intensity I_0 . If we go a step further to an inhomogeneous material, we have to consider a spatially changing attenuation coefficient $\mu(x)$. Instead of the scalar μ in the exponent of equation (2.6), we have to use a line integral over previous positions and this finally yields the law of Beer-Lambert in the form we will use:

$$I(L) = I_0 e^{-\int_0^L \mu(x) dx} \Leftrightarrow \int_0^L \mu(x) dx = -\ln(I(L)/I_0). \quad (2.7)$$

One could also include the polychromatic properties of a X-ray beam into the formulation of equation (2.7), but this is omitted at this point and will be caught up in section 6.5 concerning beam hardening. Important in the haze of equation (2.7) is to highlight that both the initial and the detected intensity values I_0 respectively $I(L)$ are accessible to measurements, quite contrary to the unknown attenuation coefficient $\mu(x)$.

2.3. Detection of X-rays

The last step before holding the projection images of the object sample in our hands is the detection of the X-rays sent out by the X-ray source. At large, the interaction of the X-rays with the detector (the specific naming of a X-ray sensor) material is just the same as with all other kind of matter as described in section 2.2. The difference is that the attenuation behaviour of the detector material is already known and this knowledge can be exploited to capture the attenuated X-ray beam. In the following step, the observed Röntgen radiation is translated into an electric signal which finally can be converted into binary coding, as used for the saving of image data. There are two diverse types of detectors: On the one hand there are the gas ionisation detectors, which are able to convert the incident X-ray beam directly into electrical energy. On the other hand there are the scintillation, solid-state detectors we will focus on hereinafter.

The mode of operation of the solid-state detectors is sketched in Figure 2.3. At first, the X-ray beam hits the layer of scintillation crystals, where the incoming radiation is converted into visible light (long-wave radiation). Behind that, a photodetector functions as an electronic light sensor. The invading light is absorbed and electrons are emitted into the photomultiplier tube using the photoelectric effect. Inside this photomultiplier tube, the electric information is amplified and aligned through multiple layers of dyodes behind the photocathode. In the end, the electrical signal is prepared to be transmitted to some digital backend devices, where it is normally converted to gray value images, the digital equivalent to a classical radiograph. Such a gray value image only contains discrete, pixelwise values which represent the average X-ray intensity for each pixel area each.

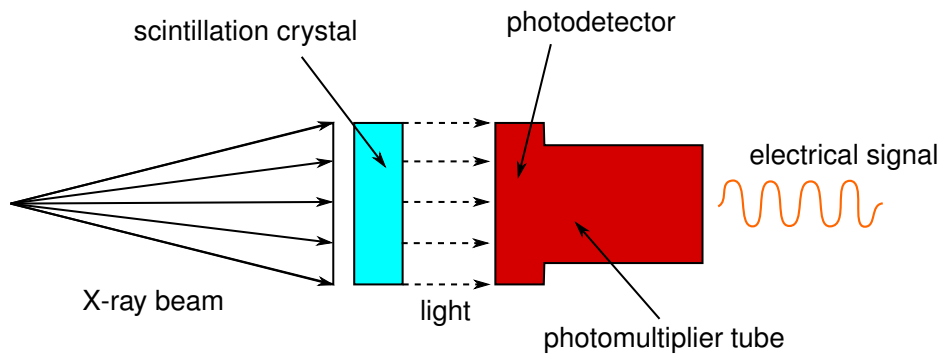


Figure 2.3.: abstracted functionality of a scintillation X-ray detector

The choice of the scintillation material is quite critical for the solid-state detectors since it has a great influence on the quality of the projection images. Some of the quality aspects determined by the choice of the material are the detection efficiency, the stability over time, the energy resolution or the intensity of afterglow phenomena. For instance, a very good efficiency of the conversion from X-rays to light pulses or a fast decay of fluorescence is needed. The detector Dexela 1512NDT¹ in use at the in-house XRCT lab of the Institute of Mechanics (CE) at the University of Stuttgart for example makes use of gadolinium oxysulfide ($\text{Gd}_2\text{O}_2\text{S}$).

¹http://www.perkinelmer.com/lab-solutions/resources/docs/PRD_CMOS_NDT_PUB_110_Rev1.pdf

3. Selected experimental scan setups

In this chapter, we will focus on a set of geometrical setups for μ XRCT scanning. The parallel beam scanning scenario in section 3.1 is rather of theoretical, preparational interest for chapter 4 (reconstruction fundamentals) than for experimental measuring, although it can also be applied to reconstruct some measurements recorded in the rare synchrotron installations. Fan beam scanning in 2D with a flat detector (see section 3.2) is the preliminary stage of the 3D cone beam scanning setup applied in most experimental μ XRCT systems, which is presented in section 3.3. A simple extension of cone beam scanning is given in section 3.4, namely helical cone beam scanning, where the rotational stage translates vertically during the projection images are taken. The final section 3.5 shortly outlines the recording process for projections and the involved coordinate system definitions which apply for the whole work.

3.1. Parallel beam scanning

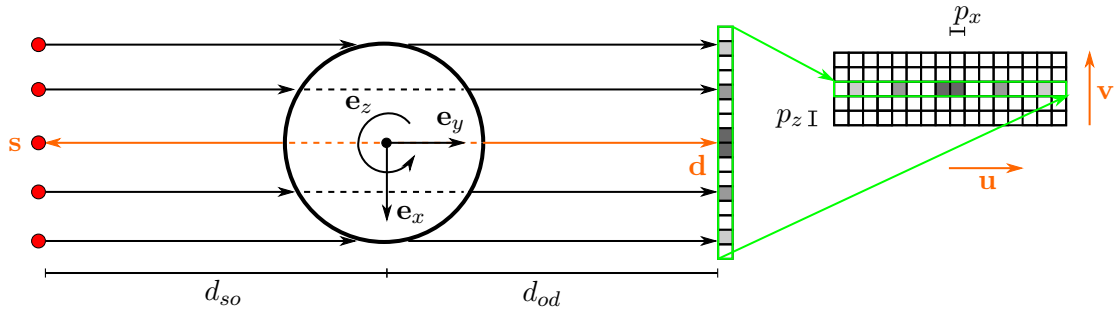


Figure 3.1.: simplified representation of a 2D or 3D parallel beam scanning scenario (the pictured vectors are only symbolic and not to be understood true-to-scale)

The first μ XRCT scanner construction one could imagine is parallel beam scanning like portrayed in Figure 3.1. The circle in the middle of the picture symbolizes the object under investigation and in its center the Cartesian coordinate system is fixed. While the e_x - e_y -plane lies in the paper plane, the object is rotated around the e_z -axis counter clockwise to obtain a set of projection images. Furthermore, the vectors s and d point from the object's center to the X-ray source respectively from the object's center to the middle pixel of the detector. Other elementary sizes are the distance from the X-ray source to the origin of the coordinate system d_{so} ($d_{so} = \|s\|_2$) and the distance from the origin of the coordinate system to the detector's center d_{od} ($d_{od} = \|d\|_2$).

The representation in the left part of Figure 3.1 shows the simplified building of parallel beam scanning in 2D. The detection of the values for the remaining X-ray intensity occurs only on a strip of

detector pixels, of which each of the N_x many pixels has the dimensions $p_x \times p_z$. In most experimental cases, the pixels of the detector are square. On top of that, parallel beam scanning is not limited to 2D, but can directly be extended to 3D, as depicted on the right in Figure 3.1. The detector consists of a rectangular grid of $N_x \times N_z$ pixels with the same dimensions as in the 2D case, $p_x \times p_z$. The pair of vectors $\{\mathbf{u}, \mathbf{v}\}$ only makes sense in this scenario and they indicate the vector from the middle pixel to the next pixel on the right in horizontal direction respectively to the next pixel at the top in vertical direction. In summary, this pair of vectors give the spatial orientation of the detector's surface related to the given Cartesian coordinate system.

3.2. Fan beam scanning

In parallel beam scanning, the assumption that all Röntgen rays are parallel to each other is quite advantageous. But parallel beams can only be achieved in rare synchrotron installations. In contrast, the fan beam scanning setup visualized in Figure 3.2 is accessible to a wider mass of users. The main difference is that the X-ray source is assumed to be similar to a point source from where the Röntgen radiation is sent out in the shape of a fan.

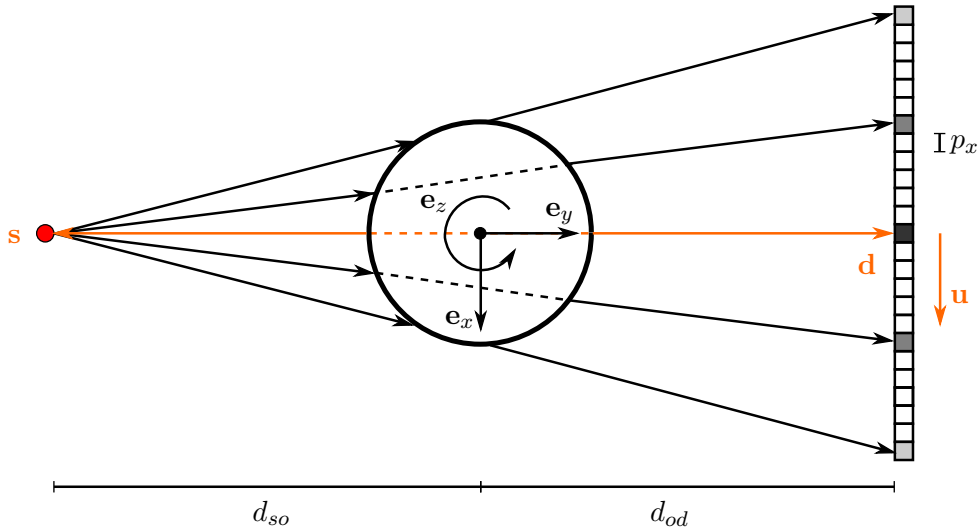


Figure 3.2.: simplified representation of a 2D fan beam scanning scenario

The pictured parameters in Figure 3.2 describing the geometric structure of the scanning situation stay almost unchanged compared to the parallel beam scanning setup in section 3.1. Obviously, sizes like the distance from the X-ray source to the origin of the coordinate system d_{so} only refer to the midline ray. As implied by the illustration, the sample material is enlarged through the so-called geometric magnification effect, which can be quantified by the size

$$M := \frac{d_{so} + d_{od}}{d_{so}}. \quad (3.1)$$

The version of fan beam scanning considered here is equal-spaced fan beam scanning for a flat detector, since the detector elements are aligned in a flat detector array with equal spacing p_x . Quite contrary to this is the idea of an equiangular fan beam geometry with a curved array of detector elements, where the angle between two neighboring rays is constant over the total opening angle and all X-ray beams have the same length.

3.3. Cone beam scanning

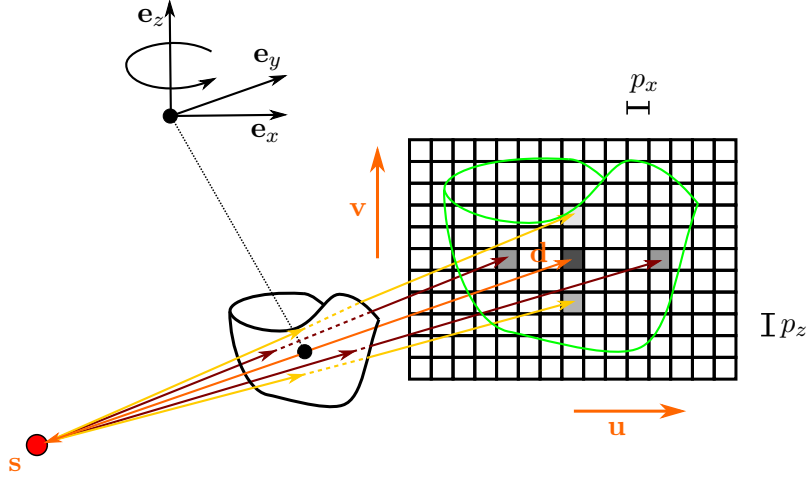


Figure 3.3.: simplified representation of a 3D cone beam scanning scenario

A further generalization of the 2D fan beam scanning scenario is the standard setup for experimental investigations: the cone beam scanning geometry. As it was also the case in section 3.1, cone beam scanning represents the direct transfer of fan beam scanning with a flat panel detector to 3D. A sketch of the resulting geometrical setup can be found in Figure 3.3. Instead of a flat fan of X-rays, the point-like Röntgen source emits X-rays inside a cone with maximal opening angle γ_m . Even if the sizes d_{so} and d_{od} are not shown in the picture, they are still given as the Euclidean norm of the vectors s and d of the middle X-ray, also referred to as optical axis in this context. The characterization of the detector surface is just the same as for parallel beam scanning. In Figure 3.3, the shade of the object in lime color on the rectangular array of detector elements clarifies the effect of the geometric magnification M from equation (3.1) in computed tomography. The geometric magnification additionally defines the resulting voxel size v_x for the depiction of the internal object structures in 3D via

$$v_x \stackrel{(3.1)}{=} \frac{p_x}{M}. \quad (3.2)$$

3.4. Helical cone beam scanning

The last step in the chain of generalizations from fan beam scanning over cone beam scanning leads us to helical cone beam scanning. Just as before, the changes of the measurement process are only minor, but this time the nature of the recorded projections differs conceptually. In the experimental setup, the object sample performs a counterclockwise rotation around the e_z -axis (as for cone beam scanning in section 3.3) and simultaneously a stepwise translation also along this axis, compare Figure 3.4. Composing these two movements yields a helical trajectory of the material under investigation in real life or of the X-ray source and detector in the imaginary mathematical description. The control of the z -translation for the specimen can for instance be quantified by the z -pitch per angular

change or per turn of 360° .

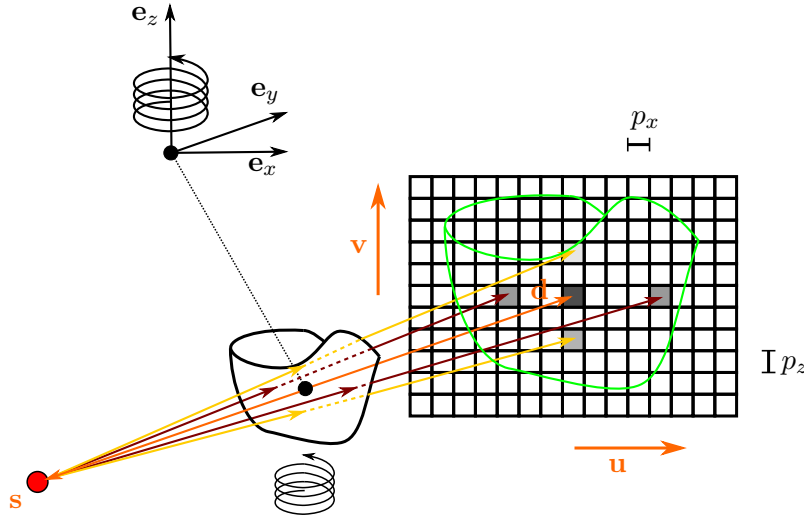
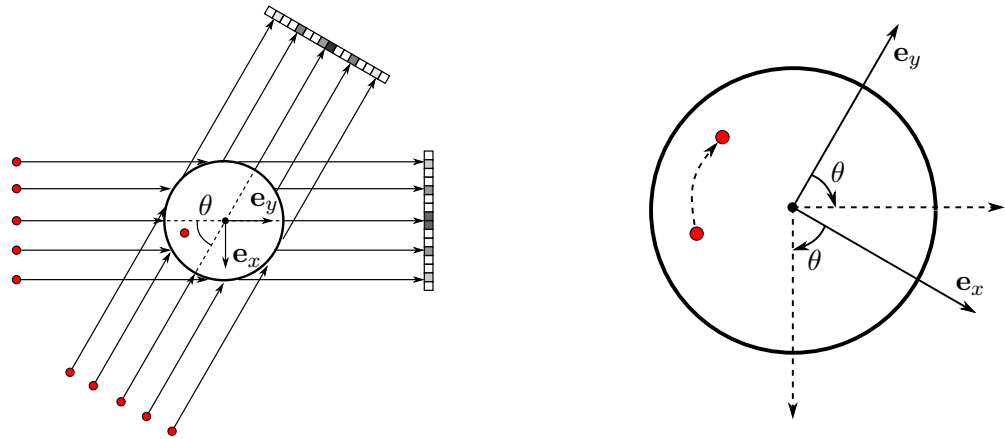


Figure 3.4.: simplified representation of a 3D helical cone beam scanning scenario

There are some reasons why helical cone beam scanning can be the method of choice despite of its increased complexity: One clear advantage lies in the fact that the limited active sensor area of the detector in vertical direction for cone beam scanning can in principle be extended to an arbitrary height, which is very useful for longitudinal samples. Especially in clinical application of CT scanning, a low radiation dose plays a central role to minimize possible effects for the patient's health - and for this reason, one single helical cone beam scan can possibly replace multiple discrete cone beam scans for the benefit of the patient in terms of the radiation dose. Another aspect at this point is the decreased exposure time for helical cone beam scanning, reducing both the radiation dose (once again) and the minor movements of the patient. The downside of helical cone beam scanning is that the reconstruction of the object's voxel data from projection data turns out to be significantly more difficult: because of the z -translation of the sample, an identical row of two different measured projection images does not refer to the same voxels of one single z -slice any more, as it is the case for cone beam scanning.

3.5. Remarks on projection acquisition and coordinate systems

We will shortly comment on the recording process of projection data for simple parallel beam scanning, as it is easily understandable for this kind of experimental setup. Figure 3.5(a), which is inspired by the book (Kak and Slaney, 2001), portrays the manner how the data acquisition will be addressed to in the mathematical description in chapter 4 as well as it will be used in the ASTRA toolbox, compare chapters 5 and 6. At large, the exact choice of the coordinate systems in all figures of chapter 3 was made like this to reproduce the description of the geometrical conditions for the ASTRA toolbox (compare van Aarle et al., 2016). The assumption herein is that the radiation source as well as the detector panel move on a circular trajectory, parametrized by the increasing projection angle θ



(a) (virtual) rotation of the X-ray source and X-ray detector

(b) actual recording mechanism corresponding to the prescribed rotation

Figure 3.5.: comparison of theoretical and empirical acquisition of projection data

between the central (parallel) X-ray beam and the e_y -axis. Of course the detected radiation intensity varies over the changing source/detector positions, as indicated by the differing gray values of the detector pixels in Figure 3.5(a). But the actual scanning process is realized experimentally by an inverse rotation of the object sample with fixed X-ray source and detector, compare Figure 3.5(b). Especially notice the contrary direction of rotation to describe the corresponding behaviour to the virtual rotation! All in all, we will use the (imaginary) recording scenario from Figure 3.5(a) in the knowledge that we can identify it with an opposite real-world scanning routine. Another remark to avoid confusions about the exact choice of the fixed Cartesian coordinate system in the center of the sample is that the whole representation of the alignment of source, object and detector is rotational invariant in the sense that the starting position of source and detector relative to the object is irrelevant. We will make use of this in sections 4.3 and 4.4, so do not hang up on this “inconsistency”.

4. Reconstruction fundamentals

One main component of the work on hand is to develop and present some important reconstruction techniques for a set of projection images. The focus of course lies on reconstruction algorithms that can be chosen inside the ASTRA toolbox (which will be introduced in chapter 5). Like a house of cards, we will gradually build up the concept of reconstruction in general as well as analytical reconstruction methods based on this derivation. The crucial question for the analytical reconstruction approach is how the local attenuation values $\mu(x, y)$, which will mostly be replaced by a more general two-dimensional object field function $f(x, y)$, can be regained from projection data. Sections 4.1 through 4.4 will cover this, and the description herein is geared to the understandable but profound explanation of the book (Buzug, 2008). In conclusion, section 4.5 elucidates algebraic reconstruction methods based on the examples of the SIRT and CGLS algorithms, whose origin is described in a completely different mathematical way.

4.1. Radon transformation and Fourier slice theorem

4.1.1. Radon transformation

The starting point for any kind of reconstruction is the mathematical description of the projection image recording process. For a fixed Cartesian coordinate system $(\mathbf{e}_x, \mathbf{e}_y)$ with given local attenuation coefficients written as $f(x, y)$ or $\mu(x, y)$, we characterize a straight line \mathbf{L} representing a single X-ray beam of a parallel beam setting via

$$\xi = x \cos(\gamma) + y \sin(\gamma). \quad (4.1)$$

To derive this, another Cartesian coordinate system $(\mathbf{e}_\xi, \mathbf{e}_\eta)$ is established, which is the rotation of the $(\mathbf{e}_x, \mathbf{e}_y)$ system by γ in mathematical positive direction as depicted in Figure 4.1(a). The meaning of any $\xi \neq 0$ for the corresponding parallel X-ray beam is a shift away from the center of both coordinate systems, while the η -variable gives the location on the path of the ray after rotating it by a fixed angle γ . Therefore, for fixed γ, ξ , the definite projection integral from 0 to the position s standing for the integration length along the affiliated ray is

$$p(s) = \int_0^s \mu(\eta) \, d\eta \quad (4.2)$$

or in a discrete version

$$p(s = K_s \Delta\eta) = \sum_{k=1}^{K_s} \mu_k \Delta\eta. \quad (4.3)$$

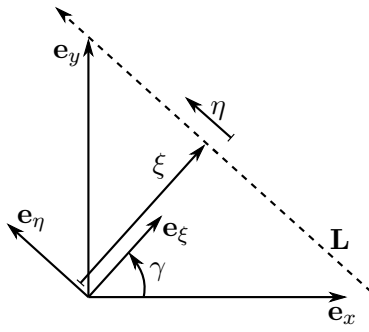
Rewriting equation (4.2) yields the definition of one single projection $p_\gamma(\xi)$ at an angle γ and a translation ξ from the origin:

$$p_\gamma(\xi) = \int_0^s \mu(\xi, \eta) \, d\eta. \quad (4.4)$$

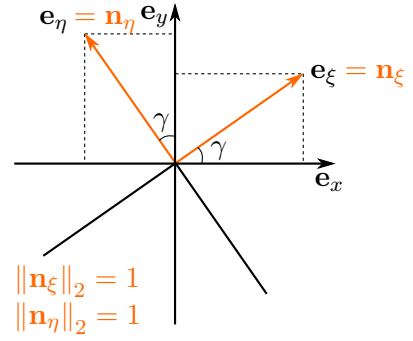
For a parallel beam scenario, the integration length s stays constant for varying ξ . In a next step, equation (4.4) shall be reformulated in the $(\mathbf{e}_x, \mathbf{e}_y)$ coordinate system. From Figure 4.1(b) it is obvious that the normal unit vectors $\mathbf{n}_\xi = (\cos(\gamma), \sin(\gamma))^T$ and $\mathbf{n}_\eta = (-\sin(\gamma), \cos(\gamma))^T$ span the rotating $(\mathbf{e}_\xi, \mathbf{e}_\eta)$ sampling coordinates. Therefore, for a single object point $\mathbf{r} = (x, y)^T$ the coordinates on a specific X-ray at fixed angle γ are given in the new coordinate system by

$$\begin{cases} \xi = \mathbf{r} \cdot \mathbf{n}_\xi = \begin{bmatrix} x \\ y \end{bmatrix} \cdot \begin{bmatrix} \cos(\gamma) \\ \sin(\gamma) \end{bmatrix} = x \cos(\gamma) + y \sin(\gamma) \\ \eta = \mathbf{r} \cdot \mathbf{n}_\eta = \begin{bmatrix} x \\ y \end{bmatrix} \cdot \begin{bmatrix} -\sin(\gamma) \\ \cos(\gamma) \end{bmatrix} = -x \sin(\gamma) + y \cos(\gamma) \end{cases} \quad (4.5)$$

and we can write $f(x, y) = \mu(\xi(x, y), \eta(x, y))$.



(a) representation of one specific X-ray in the $(\mathbf{e}_\xi, \mathbf{e}_\eta)$ coordinate system



(b) normal unit vectors \mathbf{n}_ξ and \mathbf{n}_η spanning the new coordinate system

Figure 4.1.: introduction of the rotated polar $(\mathbf{e}_\xi, \mathbf{e}_\eta)$ coordinate system

Combining all these preparational steps, we can find the two-dimensional Radon transformation of the object of interest as

$$p_\gamma(\xi) = p(\xi, \gamma) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \delta(x \cos(\gamma) + y \sin(\gamma) - \xi) dx dy \quad (4.6)$$

with the Dirac δ -function permitting only object points on the projection line \mathbf{L} in Figure 4.1(a). As a short form, we could write $f(x, y) \xrightarrow{\mathcal{R}_2} p_\gamma(\xi)$ or $p_\gamma(\xi) = \mathcal{R}_2\{f(x, y)\}$.

At this point, a useful concept can be mentioned for the first time: the sinogram as an alternative way to arrange the obtained projection images. Assuming we only have a one-dimensional detector array for a parallel beam scanning setup like in Figure 3.1 (or analogously for fan beam scanning, Figure 3.2), the sinogram displays the data in a $(\mathbf{e}_\xi, \mathbf{e}_\gamma)$ grid. The ξ -variable is the direct reference to a position on the detector array, whereas the angle $\gamma \in [0, \pi]$ (or even $[0, 2\pi]$) runs through all X-ray source and detector locations that have been headed for. If a three-dimensional scanning construction is used, we can generate a sinogram for each height in \mathbf{e}_z -direction separately (corresponding to a special row of the rectangular detector array), containing the information of a single sectional plane of the full sample. The detailed and illustrated explanation of sinograms follows in section 6.2.

4.1.2. Fourier slice theorem

The two-dimensional Radon transformation from subsection 4.1.1 is one part of the so-called Fourier slice theorem which makes computed tomography possible at all. To summarize the theorem in one sentence, one could say that the Fourier slice theorem gives the procedure to identify the one-dimensional Fourier transform of a projection with a radial line in the Cartesian Fourier space of the object data at a certain angle of measurement. We will develop the individual ingredients to the theorem one by one. If we start with some projection data $p_\gamma(\xi)$ and the reconstruction aim is to restore the local attenuation values $f(x, y)$, the main steps of the Fourier slice theorem are the following:

- (1) Calculate the one-dimensional Fourier transform $p_\gamma(\xi) \xrightarrow{\mathcal{F}_1} P_\gamma(q)$.
- (2) Construct the Fourier transformation F from P_γ : $P_\gamma(q) \xrightarrow{?} F(u, v)$.
- (3) Calculate the inverse two-dimensional Fourier transformation $F(u, v) \xrightarrow{\mathcal{F}_2^{-1}} f(x, y)$.

The steps (1) and (3) are quite straightforward: For (1), the Radon space data $p_\gamma(\xi)$ are seen as one-dimensional functions of the detector coordinate ξ for each projection angle γ . With this interpretation

$$P_\gamma(q) = P(q, \gamma) = \int_{-\infty}^{\infty} p_\gamma(\xi) e^{-i2\pi q\xi} d\xi \quad (4.7)$$

is indeed the one-dimensional Fourier transform of $p_\gamma(\xi)$ to the frequency space of projection images. On top of that, if we want to regain $f(x, y)$ from the Fourier space data $F(u, v)$ (where both spaces are Cartesian) in (3), we directly apply the inverse two-dimensional Fourier transformation to F so that

$$f(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F(u, v) e^{i2\pi(xu+yv)} du dv. \quad (4.8)$$

The sticking point in step (2) is to interconnect the differing spaces (e_q, e_γ) and (e_u, e_v) . Since (e_q, e_γ) is a polar coordinate system, a relation similar to the one in equation (4.5) has to be exploited. The interactive structure in Figure 4.2 elucidates the relations between all the introduced spaces up to now.

The transformation from the polar coordinates (e_q, e_γ) of the projection frequency space to the Cartesian coordinates (e_u, e_v) of the object frequency space follows the simple correlation

$$\begin{cases} u = q \cos(\gamma) \\ v = q \sin(\gamma) \end{cases} \quad (4.9)$$

Therewith, we declare the Fourier slice theorem as the following:

$$F(u(q, \gamma), v(q, \gamma)) = F(q \cos(\gamma), q \sin(\gamma)) = P_\gamma(q). \quad (4.10)$$

Verbatim, equation (4.10) states that a linear radial intersection of the two-dimensional Fourier spectrum $F(u, v)$ of the spatial distribution of attenuation values $f(x, y)$ at angle γ equals the one-dimensional Fourier transform $P_\gamma(q)$ of the measured Radon values $p_\gamma(\xi)$ that result from the projection of $f(x, y)$ under the angle γ . We will prove this theorem in the next paragraph.

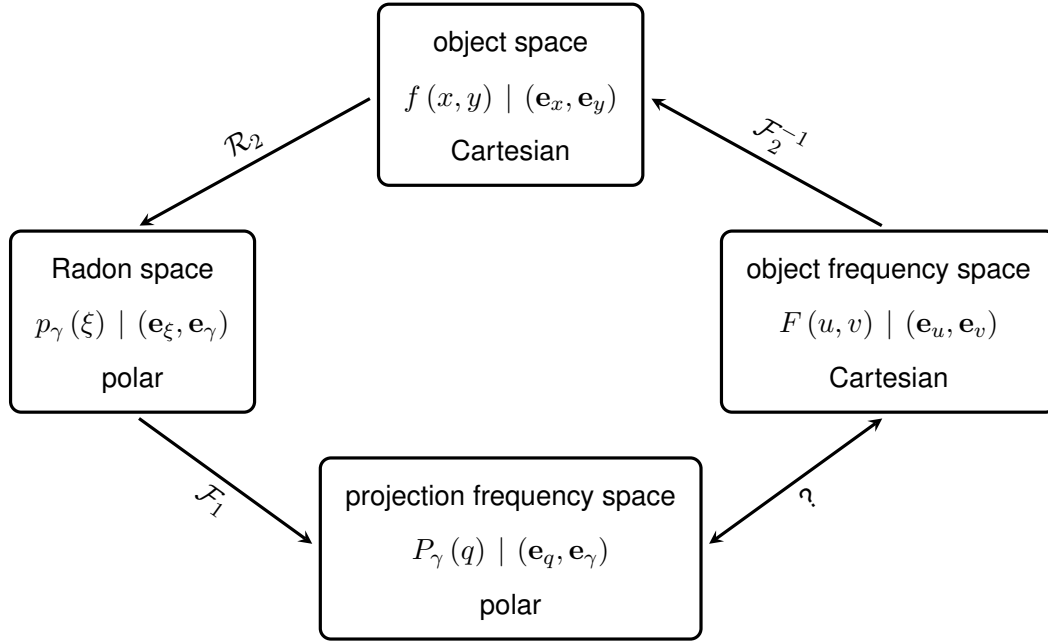


Figure 4.2.: interactive structure to clarify the coherence of the various spaces used in the discussion of the Fourier slice theorem

Let's begin to prove the equality in (4.10) starting from the right side with $P_\gamma(q)$. Looking back to equation (4.7), the one-dimensional Fourier transform of $p_\gamma(\xi)$ is

$$\begin{aligned}
 P_\gamma(q) &= \int_{-\infty}^{\infty} p_\gamma(\xi) e^{-i2\pi q\xi} d\xi \stackrel{(*)}{=} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \mu(\xi, \eta) e^{-i2\pi q\xi} d\xi d\eta \\
 &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \underbrace{\mu(\xi(x, y), \eta(x, y))}_{=f(x, y)} e^{-i2\pi q \overbrace{(\mathbf{r} \cdot \mathbf{n}_\xi)}^{=\xi}} dx dy
 \end{aligned} \tag{4.11}$$

where in the step indicated by the $(*)$ -symbol the projection (see equation (4.4))

$$p_\gamma(\xi) = \int_0^s \mu(\xi, \eta) d\eta$$

was extended to the interval $(-\infty, \infty)$. The substitutions in the last row have already been discussed in subsection 4.1.1 in equation (4.5) and below. Contrary, starting with $F(u, v)$, we can find

$$\begin{aligned}
 F(u, v) &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) e^{-i2\pi(ux+vy)} dx dy \stackrel{(**)}{=} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) e^{-i2\pi q(x \cos(\gamma) + y \sin(\gamma))} dx dy \\
 &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) e^{-i2\pi q(\mathbf{r} \cdot \mathbf{n}_\xi)} dx dy.
 \end{aligned} \tag{4.12}$$

Herein, in the step labeled with $(**)$ the relation from equation (4.9) was used to replace the (e_u, e_v) -coordinates. The final equation mark is obtained by factorisation with respect to q and finding $\xi = (\mathbf{r} \cdot \mathbf{n}_\xi)$ following equation (4.5) once again. Altogether, the results of equation (4.11) and (4.12) are identical, which was to be proved.

The shortest description of the Fourier slice theorem can be specified by

$$F(u, v)|_{u=q \cos(\gamma), v=q \sin(\gamma)} = P_\gamma(q). \quad (4.13)$$

For the sake of overview, a concluding flow chart can be found in Figure 4.3.

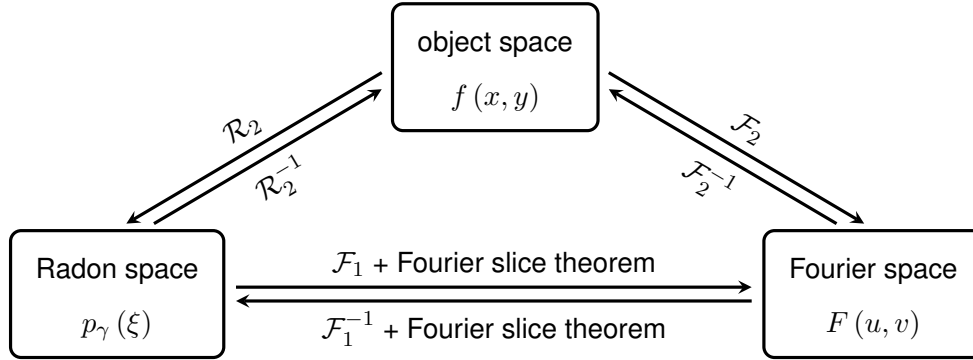


Figure 4.3.: flow chart to emphasize the overall benefit of the Fourier slice theorem as a linking element in computed tomography

4.1.3. Remarks and challenges

With the help of the Fourier slice theorem (4.13), the algorithmic scheme to reconstruct the original sample of interest arises, like already suggested in Figure 4.3. In the beginning, the only resource we have in our hands is the measured projection data $p_\gamma(\xi)$ living in the Radon space (e_ξ, e_γ) . After applying the one-dimensional Fourier transform \mathcal{F}_1 , the projection data translate into polar spectral data $P_\gamma(q)$ in the Fourier-transformed Radon space (e_q, e_γ) . The connection to the Cartesian Fourier space is given by the fact that for a fixed angle γ , $P_\gamma(q)$ represents a line through the origin in the (e_u, e_v) Fourier space under the angle γ with the e_u -axis. Including the whole set of possible angles leads to a radial filling of the Fourier space, as depicted in Figure 4.4 by the solid green lines.

Before we can use the inverse two-dimensional Fourier transform \mathcal{F}_2^{-1} for the Fourier space data $F(u, v)$, it is unavoidable to sort the line information of $P_\gamma(q)$ into the Cartesian grid (u, v) . This procedure is also called Cartesian regridding and is necessary since a numeric implementation of any kind of Fourier transformation (e.g. the fast Fourier transform, FFT) is only possible for a regular rectangular grid just like (e_u, e_v) . If this has happened, we can restore the tomographic data $f(x, y)$ in its object space (e_x, e_y) . At this point, the so called forward projection is able to close the circle to some projection data $p_\gamma(\xi)$, which was our starting point. Forward projecting only means to use the Radon transform \mathcal{R}_2 for given object data $f(x, y)$ (which is more of theoretical interest than of practical).

The challenge of reconstruction algorithms becomes clear if we switch from the analytical to the experimental point of view: We only have a finite number of projections at angles γ_i available for reconstruction. That means there is only a limited number of measured radial lines in the e_u - e_v -plane, and even more only a set of discrete points on these lines, displaying the single detector pixels. The measured projections for increasing angle γ_i are symbolized by the dashed green lines in Figure 4.4 and the discrete detector pixel values are given by the orange points on these lines. As already mentioned, we need values on the Cartesian grid points (pictured by the blank points) to realize a

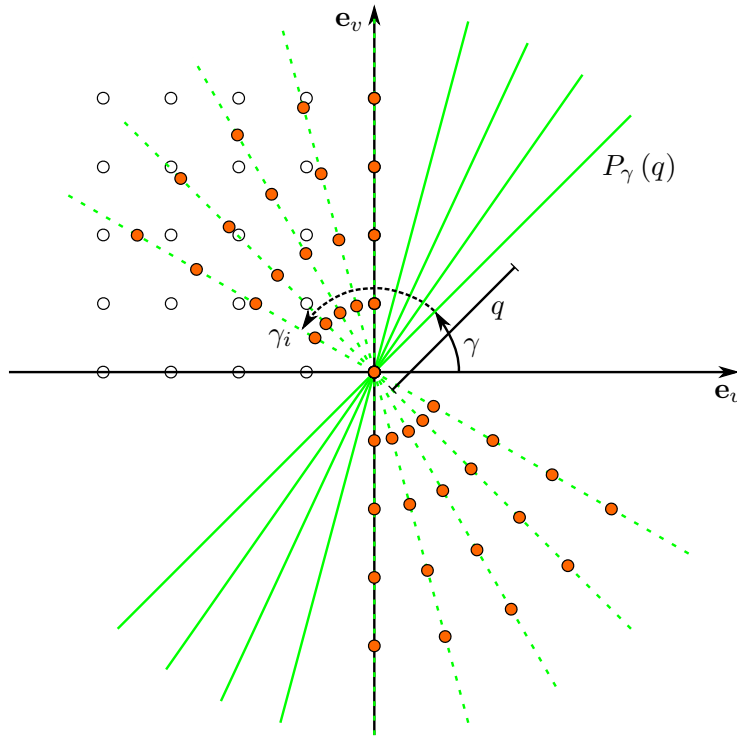


Figure 4.4.: continuous and discrete spectral projection data in two competitive coordinate systems

numerical Fourier transformation operation, so that the detector pixel values have to be interpolated onto this grid. Firstly, this interpolation is not trivial because of the radial arrangement of the point cloud (proposing methods like nearest neighborhood search or bilinear interpolation). Another important phenomenon that can be observed in Figure 4.4 is that the density of points decreases for higher spatial frequencies. Therefore, interpolating in remote regions from the origin yields a much higher interpolation error and related uncertainty. This finally leads to a severe degradation of the reconstruction image quality especially for details, since they have a high spatial frequency nature.

4.2. 2D reconstruction methods for parallel beam scanning

After the general introduction into how reconstruction can be formulated mathematically in section 4.1, we will devote ourselves to reconstruction methods for two-dimensional parallel beam scanning. What was not mentioned in the small paragraph on sinograms in subsection 4.1.1 is the fact that the (mathematically) total necessary projection angle range differs for varying scanning scenarios. For the parallel beam scanning scenario, the interval $[0, \pi]$ is sufficient because the measured attenuation coefficients replicate after half a turn. This lies in the nature of parallel beam scanning: the parallel X-ray beams tread identical paths as before after a rotation of 180° . For fan or cone beam scanning in sections 4.3 and 4.4 though, this is not the case - here, the full range of $[0, 2\pi]$ for the projection angles is required.

4.2.1. Simple backprojection

The first real reconstruction algorithm presented here is the simple backprojection algorithm. As we will see, the simple backprojection only serves as a preparation and motivation for the filtered backprojection algorithm in subsection 4.2.2. The basic idea of backprojection throughout lies in smearing back the measured projection profiles $p_\gamma(\xi)$ in source direction. In mathematical terms,

$$g(x, y) = \int_0^\pi p_\gamma(\xi) d\gamma \stackrel{(4.1)}{=} \int_0^\pi p_\gamma(x \cos(\gamma) + y \sin(\gamma)) d\gamma \quad (4.14)$$

gives the reconstructed attenuation coefficient at the position $\mathbf{r} = (x, y)^T$. The arising problem of the smear back operation is that each point in the reconstruction image grid receives a non-negative attenuation contribution from all other points of the original image, even if a grid point is actually located outside the real sample. The reason for this is explained in the following paragraph.

Let's investigate the reconstructed attenuation value from equation (4.14) deeper (herein, \mathbf{L} is a specific (back)projection line, see Figure 4.1(a)):

$$\begin{aligned} g(x, y) &= \int_0^\pi \overbrace{\iint_{\mathbf{r} \in \mathbf{L}} f(\mathbf{r}) d\mathbf{r}}^{=p_\gamma(\xi)} d\gamma = \int_0^\pi \iint_{\mathbf{r} \in \mathbb{R}^2} f(\mathbf{r}) \delta(\mathbf{r} - \mathbf{L}) d\mathbf{r} d\gamma \\ &= \int_0^\pi \iint_{\mathbf{r} \in \mathbb{R}^2} f(\mathbf{r}) \delta((\mathbf{r} \cdot \mathbf{n}_\xi) - \xi) d\mathbf{r} d\gamma \stackrel{(*)}{=} \int_0^\pi \iint_{\mathbf{r}' \in \mathbb{R}^2} f(\mathbf{r}') \delta((\mathbf{r} \cdot \mathbf{n}_\xi) - (\mathbf{r}' \cdot \mathbf{n}_\xi)) d\mathbf{r}' d\gamma \quad (4.15) \\ &= \iint_{\mathbf{r}' \in \mathbb{R}^2} f(\mathbf{r}') \left[\int_0^\pi \delta((\mathbf{r} - \mathbf{r}') \cdot \mathbf{n}_\xi) d\gamma \right] d\mathbf{r}'. \end{aligned}$$

In the steps indicated with $(*)$ in equations (4.15) and (4.16), we use some properties of the Dirac δ -function which will not be explained any further in here. These characteristics can for example be looked up in (Buzug, 2008, p. 175 ff.). After having defined φ as the angle between the vector $\mathbf{r} - \mathbf{r}'$ and the \mathbf{e}_x -axis of the Cartesian coordinate system, we can continue with the reformulation of equation (4.15) by replacing $(\mathbf{r} - \mathbf{r}') \cdot \mathbf{n}_\xi$ with $\|\mathbf{r} - \mathbf{r}'\|_2 \cos(\varphi - \gamma)$:

$$\begin{aligned} g(x, y) &= \iint_{\mathbf{r}' \in \mathbb{R}^2} f(\mathbf{r}') \left[\int_0^\pi \delta(\|\mathbf{r} - \mathbf{r}'\|_2 \cos(\varphi - \gamma)) d\gamma \right] d\mathbf{r}' \\ &\stackrel{(*)}{=} \iint_{\mathbf{r}' \in \mathbb{R}^2} f(\mathbf{r}') \left[\int_0^\pi \frac{\delta(\gamma - \gamma_0)}{\|\mathbf{r} - \mathbf{r}'\|_2 \underbrace{|\sin(\pm\pi/2)|}_{=1}} d\gamma \right] d\mathbf{r}'. \quad (4.16) \end{aligned}$$

The new size γ_0 is the only zero value of $\cos(\varphi - \gamma)$ for $\gamma \in [0, \pi)$. Because the denominator term $\|\mathbf{r} - \mathbf{r}'\|_2$ is independent of the integrator and $\int_0^\pi \delta(\gamma - \gamma_0) d\gamma = 1$, we find

$$g(x, y) = \iint_{\mathbf{r}' \in \mathbb{R}^2} f(\mathbf{r}') \frac{1}{\|\mathbf{r} - \mathbf{r}'\|_2} d\mathbf{r}' = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x', y') \frac{1}{\|(x - x', y - y')^T\|_2} dx' dy'. \quad (4.17)$$

Equation (4.17) exactly equals the convolution (integral) of the object data $f(x, y)$ with the so called point-spread function (PSF) or also impulse response $h(x, y) = 1/\|(x, y)^T\|_2$. In short version, we

could write

$$g(x, y) = f(x, y) * h(x, y) \quad (4.18)$$

for the whole simple backprojection routine. If we consider only a single object point $f(x, y) = \delta(x, y)$ present, the reconstruction algorithm for simple backprojection smears back the attenuation coefficient with the point-spread function. The attenuation value therefore influences the neighboring region with the radial decreasing factor $1/\|\mathbf{r}\|_2$. But since this is just the influence of one sole point information, we obtain an overall unacceptable blurring of the reconstruction image - and so the need for other algorithmic ideas becomes clear.

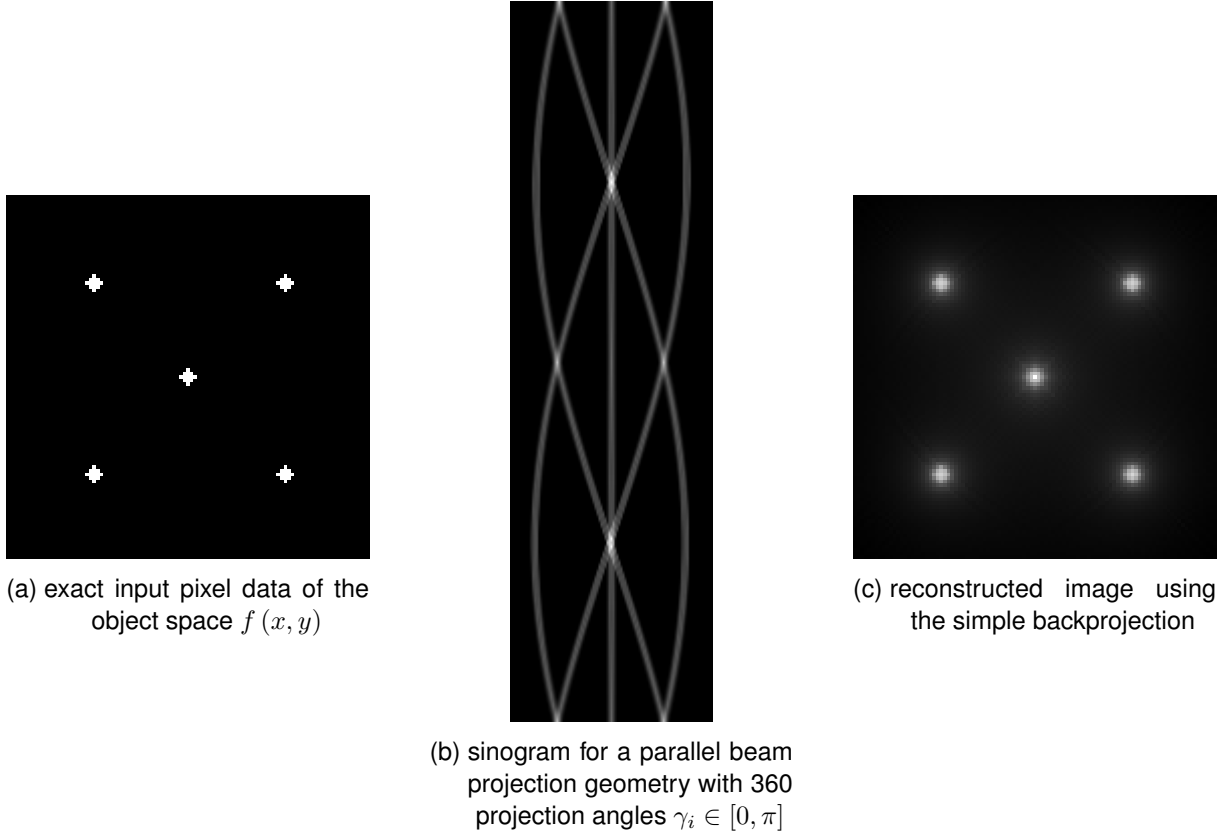


Figure 4.5.: blurring introduced by the usage of the ASTRA simple backprojection algorithm BP

For the purpose of clarification, Figure 4.5 shows the effect of a reconstruction trial applying the simple backprojection algorithm BP from the ASTRA toolbox (for details consider chapter 5). The five point-like bright objects of the exact input slice in Figure 4.5(a) are first reshaped into the sinogram in Figure 4.5(b) using a (simulated) parallel beam scanning setup. The ASTRA equivalent to simple backprojection, the BP algorithm, takes this sinogram as starting point for the reconstruction and results in the image displayed in Figure 4.5(c). The radial filling around the point-like objects and the overall blurring are clearly recognizable.

4.2.2. Filtered backprojection

As we have already seen in section 4.1 (e.g. (2) in subsection 4.1.2), the sticking point of reconstructing/backprojecting is the change from Cartesian to polar coordinates and vice versa. This is the starting point for the derivation of the filtered backprojection reconstruction algorithm. We recapitulate equation (4.8), the inverse two-dimensional Fourier transformation of some Fourier space

data $F(u, v)$:

$$f(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F(u, v) e^{i2\pi(xu+yv)} du dv.$$

To express this relation in the (e_q, e_γ) polar coordinate system, equation (4.9) serves as a transformation relationship. But this time, we take a closer look at the coordinate transform for the line elements which act as integrators. We are interested in the mapping $du dv \mapsto J dq d\gamma$ with the Jacobian J given by

$$J = \begin{vmatrix} \frac{\partial u}{\partial q} & \frac{\partial u}{\partial \gamma} \\ \frac{\partial v}{\partial q} & \frac{\partial v}{\partial \gamma} \end{vmatrix} = \begin{vmatrix} \cos(\gamma) & -q \sin(\gamma) \\ \sin(\gamma) & q \cos(\gamma) \end{vmatrix} = q \underbrace{(\cos(\gamma)^2 + \sin(\gamma)^2)}_{=1} = q. \quad (4.19)$$

If we substitute the coordinate variables in equation (4.8) with the coordinate transform in equation (4.19), this results in

$$f(x, y) = \int_0^{2\pi} \int_0^{\infty} \underbrace{F(q \cos(\gamma), q \sin(\gamma))}_{=F(q, \gamma)} e^{i2\pi q(x \cos(\gamma) + y \sin(\gamma))} q dq d\gamma. \quad (4.20)$$

In the next step, the outer integral is split into the two integration intervals $[0, \pi]$ and $[\pi, 2\pi]$, which yields

$$\begin{aligned} f(x, y) &= \int_0^{\pi} \int_0^{\infty} F(q, \gamma) e^{i2\pi q(x \cos(\gamma) + y \sin(\gamma))} q dq d\gamma \\ &\quad + \int_{\pi}^{2\pi} \int_0^{\infty} F(q, \gamma + \pi) e^{i2\pi q(x \cos(\gamma + \pi) + y \sin(\gamma + \pi))} q dq d\gamma. \end{aligned} \quad (4.21)$$

The terms $\cos(\gamma + \pi)$ and $\sin(\gamma + \pi)$ are equal to $-\cos(\gamma)$ respectively $-\sin(\gamma)$. An additional set of symmetry properties for the Fourier transform is necessary for the further handling of equation (4.21) (compare Klingen, 2001, p. 184 ff.):

$$\Re F(q, \gamma) = \Re F(-q, \gamma + \pi) = \Re F(-q, \gamma) = \Re F(q, \gamma + \pi) \quad (4.22)$$

$$\Im F(q, \gamma) = \Im F(-q, \gamma + \pi) = -\Im F(-q, \gamma) = -\Im F(q, \gamma + \pi). \quad (4.23)$$

Therewith, we write

$$\begin{aligned} f(x, y) &= \int_0^{\pi} \int_0^{\infty} (\Re F(q, \gamma) + i \Im F(q, \gamma)) e^{i2\pi q(x \cos(\gamma) + y \sin(\gamma))} q dq d\gamma \\ &\quad + \int_0^{\pi} \int_0^{\infty} \left(\overbrace{\Re F(q, \gamma)}^{=\Re F(q, \gamma + \pi)} - i \overbrace{\Im F(q, \gamma)}^{=-\Im F(q, \gamma + \pi)} \right) e^{-i2\pi q(x \cos(\gamma) + y \sin(\gamma))} q dq d\gamma \end{aligned} \quad (4.24)$$

$$\begin{aligned} &= \int_0^{\pi} \int_0^{\infty} (\Re F(q, \gamma) + i \Im F(q, \gamma)) e^{i2\pi q(x \cos(\gamma) + y \sin(\gamma))} q dq d\gamma \\ &\quad + \int_0^{\pi} \int_{-\infty}^0 (\Re F(q, \gamma) + i \Im F(q, \gamma)) e^{i2\pi q(x \cos(\gamma) + y \sin(\gamma))} (-q) dq d\gamma \end{aligned} \quad (4.25)$$

The difference in the underbraced part of equation (4.24) is the switch of the limits of integration, changing the sign at the positions marked in green. To come from the underbraced part of equation (4.24) to equation (4.25), it is necessary to use the symmetry relations in equations (4.22) and (4.23). This was also used to generate the second summand of equation (4.24), as indicated by the overbraces. Except for the red factors, both summands in equation (4.25) could be reunited. This problem can be bypassed by simply introducing the absolute value:

$$f(x, y) = \int_0^\pi \int_{-\infty}^\infty F(q, \gamma) e^{i2\pi q(x \cos(\gamma) + y \sin(\gamma))} |q| dq d\gamma \quad (4.26)$$

$$\stackrel{(4.1)}{\underset{FST}{=}} \int_0^\pi \int_{-\infty}^\infty P_\gamma(q) e^{i2\pi q\xi} |q| dq d\gamma. \quad (4.27)$$

To produce equation (4.27), the short form $F(q, \gamma) = F(q \cos(\gamma), q \sin(\gamma))$ is identified with $P_\gamma(q)$ by the means of the Fourier slice theorem (FST). Writing

$$f(x, y) = \int_0^\pi \underbrace{\left[\int_{-\infty}^\infty P_\gamma(q) |q| e^{i2\pi q\xi} dq \right]}_{=: h_\gamma(\xi)} d\gamma, \quad (4.28)$$

the reconstruction algorithm for filtered backprojection is complete. The function $h_\gamma(\xi)$ symbolizes the high-pass filtered version of the projection profile $p_\gamma(\xi)$, which of course solves the key problem of simple backprojection in subsection 4.2.1.

Summarizing the steps from the available projection data $p_\gamma(\xi)$ to the reconstructed local attenuation values $f(x, y)$ using a filtered backprojection, we can capture the three main steps in the following.

- (1) Calculate the one-dimensional Fourier transform $p_\gamma(\xi) \xrightarrow{\mathcal{F}_1} P_\gamma(q)$.
- (2) Calculate the inverse one-dimensional Fourier transform h_γ of the high-pass filtered P_γ : $|q|P_\gamma(q) \xrightarrow{\mathcal{F}_1^{-1}} h_\gamma(\xi)$.
- (3) Calculate the backprojection f along the line $\xi = x \cos(\gamma) + y \sin(\gamma)$:
 $f(x, y) = \int_0^\pi h_\gamma(\xi) d\gamma$.

4.2.3. Aspects of discretization and implementation

After we found the filtered backprojection as a practical meaningful method to reconstruct a set of measured projection images, we will shortly discuss how to make the mathematical terms of the previous subsection accessible to a computer. Let's introduce the discretization $\Delta\xi := 1/2Q$ for a weighting interval $[-Q, Q]$ of the ramp function $|q|$ if the maximum available spatial frequency shall be bounded by Q . Of course $\Delta\xi$ refers to the width of a single detector pixel. Therewith, the sampled projection signal is written as $p_\gamma(j\Delta\xi)$ for $j = 0, \dots, D-1$, where D denotes the total number of horizontal detector elements contained in the detector array. Consequently, we can restrict the integration limits of equation (4.7) to

$$P_\gamma(q) = \int_0^{(D-1)\Delta\xi} p_\gamma(\xi) e^{-i2\pi q\xi} d\xi$$

and employ a discretization in the form

$$P_\gamma \left(k \frac{2Q}{D} \right) = \frac{1}{2Q} \sum_{j=0}^{D-1} p_\gamma \left(\frac{j}{2Q} \right) e^{-i2\pi \frac{jk}{D}} \quad \text{for } k = 0, \dots, D-1 \quad (4.29)$$

with renamings $\xi \mapsto j\Delta\xi = \frac{j}{2Q}$, $q \mapsto k\Delta q := k\frac{2Q}{D}$. One obviously obtains $q\xi = k\frac{2Q}{D}j\frac{1}{2Q} = \frac{jk}{D}$ for the exponent of the exponential function, completing the expressions in equation (4.29). Analogously, the filtered projection signal $h_\gamma(\xi)$ from equation (4.28) can be windowed to

$$h_\gamma(\xi) \approx \int_{-Q}^Q P_\gamma(q) |q| e^{i2\pi q\xi} dq$$

and discretized to

$$h_\gamma \left(\frac{j}{2Q} \right) \approx \frac{2Q}{D} \sum_{k=-D/2}^{D/2} P_\gamma \left(k \frac{2Q}{D} \right) \overbrace{\left| k \frac{2Q}{D} \right|}^{(*)} e^{i2\pi \frac{jk}{D}} \quad \text{for } j = 0, \dots, D-1. \quad (4.30)$$

Concluding, the local image value in a discretized form is given by

$$f(x, y) = \int_0^\pi h_\gamma(\xi) d\gamma \approx \frac{\pi}{N_p} \sum_{n=1}^{N_p} h_{\gamma_n} (x \cos(\gamma_n) + y \sin(\gamma_n)), \quad (4.31)$$

where of course $\{\gamma_n\}_{n=1}^{N_p}$ is the set of the N_p many measuring angles at whom the projections were taken and h_{γ_n} has to be used according to equation (4.30). The prefactor π/N_p comes from the discretization of the angle $\gamma \mapsto \Delta\gamma := \pi/N_p$.

The particularity of the with $(*)$ marked term (the discretized ramp filter) in equation (4.30) is that it is a linear increasing function in the frequency domain and therefore increases the noise in the high frequency regions. That means a data windowing as already implied by the integration limits $[-Q, Q]$ is definitely necessary. Such a windowing can be carried out either in the frequency or in the spatial domain. We will just take a look at the restriction of the reconstruction routine to a rectangular band in the frequency interval $[-Q, Q]$, which can be expressed via

$$h_\gamma \left(\frac{j}{2Q} \right) \approx \frac{2Q}{D} \sum_{k=-D/2}^{D/2} P_\gamma \left(k \frac{2Q}{D} \right) \left| k \frac{2Q}{D} \right| W \left(k \frac{2Q}{D} \right) e^{i2\pi \frac{jk}{D}} \quad (4.32)$$

using a window function $W(\bullet)$. Common window functions are the Ram-Lak filter (short form for Ramachandran and Lakshminarayanan; Ramachandran and Lakshminarayanan, 1971) or the Shepp-Logan filter (Shepp and Logan, 1974):

$$W_{RL}(q) = \text{rect}(q) = \begin{cases} 1 & \text{if } |q| \leq \frac{1}{2} \\ 0 & \text{else} \end{cases} \quad (4.33)$$

$$W_{SL}(q) = \text{rect}(q) \text{sinc}(q) \quad \text{with } \text{sinc}(\bullet) = \frac{\sin(\pi\bullet)}{\pi\bullet}. \quad (4.34)$$

As suggested by the definition of the rectangular function $\text{rect}(\bullet)$, the range of nonzero values of the window function is limited to a frequency domain with $Q = 1/2$ (which could of course be expanded to any other interval by linear coordinate transformation). The difference between the Ram-Lak filter

(4.33) and the Shepp-Logan filter (4.34) is how sharp the edges of the rectangle are shaped. The sinc-factor of the Shepp-Logan window function tries to smoothen the edges and is only one of a variety of other kernels developed on top of the simple windowing ansatz by Ram-Lak. All these possible kernels can also be expressed in the spatial domain (which is of minor interest here) and so finally the reconstruction can be done numerically avoiding the problem of additional noise evoked by the ramp filter term (*).

4.3. 2D reconstruction methods for fan beam scanning

In section 3.2, 2D fan beam scanning was already scanned briefly. Especially the sizes d_{so} (distance source \leftrightarrow origin), d_{od} (distance origin \leftrightarrow detector) and their sum $d_{sd} = d_{so} + d_{od}$ (distance source \leftrightarrow detector) have been motivated in Figure 3.2. Similarly, the difference between a curved and a plane detector array has been pointed out: while in the curved variant there are equidistant angles $\Delta\zeta$ in between the X-rays leading to neighboring detector elements, the detector spacing $\Delta\xi$ for the flat variant is equidistant. For the following, we parametrize the trajectory of the X-ray source moving on a circle around the origin of the Cartesian (e_x, e_y) coordinate system in the mathematical positive sense by $s = (-d_{so} \sin(\theta), d_{so} \cos(\theta))^T$. Therein, the angle θ is included between the central beam of the X-ray beam fan and the e_y -axis, and the X-ray source starts its rotation in the point $(0, d_{so})^T$ (cf. Figure 4.6). All these sizes will be of importance for the handling of reconstruction for a fan beam scanning scenario.

One clear result of subsection 4.2.1 was that simple backprojection is an insufficient method to reconstruct projection data. For this reason, we will directly enter into the preparational steps for filtered backprojection. We have to understand how projection images taken under a fan beam scanning setup are linked to projections from parallel beam scanning. In fact, the methodology to achieve this conversion is quite easily understandable and known as rebinning. The beams of different fan beam projection angles can partially be rebinned (meaning kind of a replacement) into a set of parallel beams for one single projection angle. After this procedure, we can fall back to the reconstruction routines of section 4.2 for the newly synthesized parallel beam scanning scenario.

To approach the rebinning for a curved detector, Figure 4.6 displays the geometrical considerations which are necessary to give the following mathematical relations. The arc angle ψ belonging to a sole ray of the fan beam establishes the traveled arc length $\zeta = d_{sd}\psi$ per definitionem of a radian size. The projection angle θ is usefully defined between the central beam of the fan and the e_y -axis of the Cartesian coordinate system (e_x, e_y) . The new fan beam polar coordinates are therefore given by (e_ζ, e_θ) in contrast to the parallel beam polar coordinates (e_ξ, e_γ) (for the Radon space). While the projection data $p_\gamma(\xi)$ for parallel beam stay unchanged, we identify the projection data in a fan beam scanning setting straightforwardly by $\phi_\theta(\zeta)$. With a sharp look at Figure 4.6, the relations

$$\xi = d_{so} \sin(\psi) \quad \Leftrightarrow \quad \psi = \arcsin(\xi/d_{so}) \quad (4.35)$$

$$\gamma = \theta + \psi \quad \Leftrightarrow \quad \theta = \gamma - \psi \quad (4.36)$$

become clear. This directly yields the rebinning for fan beam scanning with a curved detector array as

$$\phi_\theta(\zeta) = \phi_\theta(d_{sd}\psi) = p_{\theta+\psi}(d_{so} \sin(\psi)) \quad (4.37)$$

$$\Leftrightarrow p_\gamma(\xi) = \phi_{\gamma-\psi}(\zeta) = \phi_{\gamma-\arcsin(\xi/d_{so})}(d_{sd} \arcsin(\xi/d_{so})). \quad (4.38)$$

What can be seen in the argument of $p_{\theta+\psi}$ in equation (4.37) is that the parallel beam data points of the rebinned projection ϕ_θ along the e_ξ -axis are not distributed equidistantly (due to the $\sin(\psi)$ -term),

inducing the need to interpolate to achieve equidistant parallel beam data.

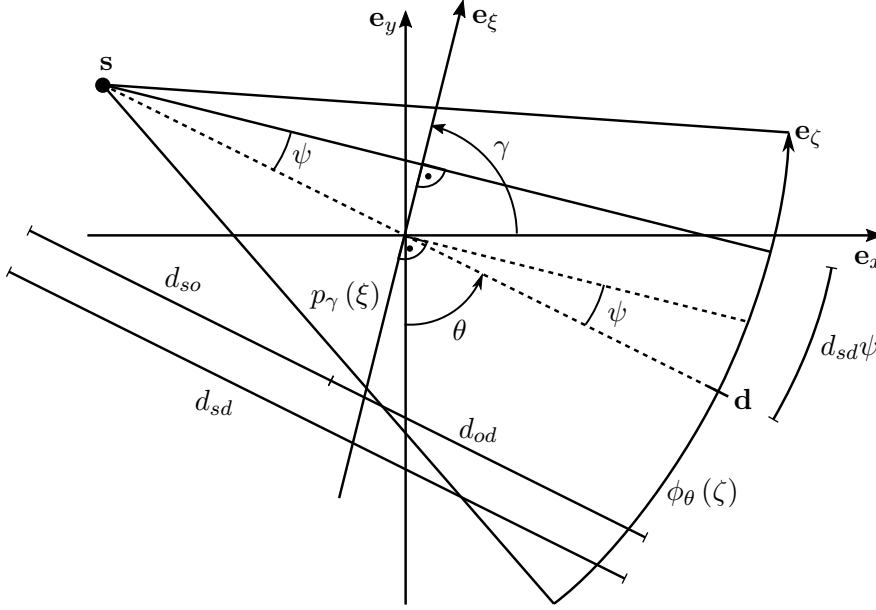


Figure 4.6.: geometrical situation for rebinning of a curved detector array

The next step towards filtered backprojection for fan beam scanning with a flat detector is to transmit the results of rebinning for a curved detector array to the planar one. This time the spacing of the detector elements is equidistant, implying a non-linear change of the angle between neighboring X-rays. Another big difference for rebinning of flat detector panels is that an imaginary linear detector going through the origin has to be used as an auxiliary, compare the e_a -axis in Figure 4.7.

The currently considered point $\mathbf{r} = (x, y)^T$ is parameterized by the radius r from the origin and the angle δ to the e_x -axis. Its distance to the radiation source s is denoted by L . The virtual detector used to derive the filtered backprojection method for fan beam scanning with a flat detector array in the following is orientated along the e_a -axis, so the polar fan beam coordinate system for planar detectors is (e_a, e_θ) . The value a (in green) designates the intersection of the e_a -axis with the actual X-ray \mathbf{L} . The meaning of the projection angle θ or the fan beam projection data $\phi_\theta(a)$ remains unchanged on the whole. We can again identify some important preparational relations by detailed contemplation of Figure 4.7:

$$\xi = a \cos(\psi) = a \frac{d_{so}}{\sqrt{a^2 + d_{so}^2}} \quad (4.39)$$

$$\gamma = \theta + \psi = \theta + \arctan(a/d_{so}) \quad (4.40)$$

$$\mathbf{r} \cdot \mathbf{n}_\xi = \xi = r \cos(\gamma - \delta). \quad (4.41)$$

record

$$f(x, y) = \frac{1}{2} \int_0^{2\pi} \left(\int_{-\infty}^{\infty} p_\gamma(\xi') g(\mathbf{r} \cdot \mathbf{n}_\xi - \xi') d\xi' \right) d\gamma \quad (4.44)$$

as an interim result.

The next level consists of the coordinate transform from the parallel beam ($\mathbf{e}_\xi, \mathbf{e}_\gamma$) system to the fan beam ($\mathbf{e}_a, \mathbf{e}_\theta$) system, for which we will excessively use the correlations introduced in equations (4.39) through (4.41). We are interested in the conversion $d\xi d\gamma \mapsto J da d\theta$ with

$$J = \begin{vmatrix} \frac{\partial \xi}{\partial a} & \frac{\partial \xi}{\partial \theta} \\ \frac{\partial \gamma}{\partial a} & \frac{\partial \gamma}{\partial \theta} \end{vmatrix} = \begin{vmatrix} \left(\frac{d_{so}}{\sqrt{a^2 + d_{so}^2}} \right)^3 & 0 \\ \frac{d_{so}}{a^2 + d_{so}^2} & 1 \end{vmatrix} = \left(\frac{d_{so}}{\sqrt{a^2 + d_{so}^2}} \right)^3 \stackrel{(4.39)}{=} \cos(\psi)^3 \quad (4.45)$$

using the auxiliary calculations

$$\begin{aligned} \frac{\partial \xi}{\partial a} &\stackrel{(4.39)}{=} \frac{\partial \left(a d_{so} \cdot (a^2 + d_{so}^2)^{-\frac{1}{2}} \right)}{\partial a} = d_{so} (a^2 + d_{so}^2)^{-\frac{1}{2}} + a d_{so} \left(-\frac{1}{2} (a^2 + d_{so}^2)^{-\frac{3}{2}} 2a \right) \\ &= d_{so} (a^2 + d_{so}^2)^{-\frac{3}{2}} - \frac{d_{so} a^2 (a^2 + d_{so}^2)^{-\frac{3}{2}}}{a^2 + d_{so}^2} = \left(\frac{d_{so}}{\sqrt{a^2 + d_{so}^2}} \right)^3 \quad \text{and} \\ \frac{\partial \gamma}{\partial a} &\stackrel{(4.40)}{=} \frac{\partial \left(\theta + \arctan \left(\frac{a}{d_{so}} \right) \right)}{\partial a} = \frac{d_{so}}{a^2 + d_{so}^2}. \end{aligned}$$

Applying the found coordinate transform condition and the set of equations (4.39) - (4.41) reshapes equation (4.44) to

$$\begin{aligned} f(r, \delta) &= \frac{1}{2} \int_0^{2\pi} \left\{ \int_{-a_{min}}^{a_{max}} \overbrace{p_{\theta+\psi}}^{=\phi_\theta(a)} \underbrace{\left(a \frac{d_{so}}{\sqrt{a^2 + d_{so}^2}} \right)}_{=\xi} \cdots \right. \\ &\quad \left. g \left(\overbrace{r \cos \left(\underbrace{\theta + \arctan \left(\frac{a}{d_{so}} \right) - \delta}_{=\gamma} \right)}^{=\mathbf{r} \cdot \mathbf{n}_\xi - \xi'} - \underbrace{a \frac{d_{so}}{\sqrt{a^2 + d_{so}^2}}}_{=\xi} \right) \underbrace{\left(\frac{d_{so}}{\sqrt{a^2 + d_{so}^2}} \right)^3}_{=J} da \right\} d\theta. \end{aligned} \quad (4.46)$$

In order to shorten the large expressions, we have to insert some mathematical side thoughts. At first, we will use an addition theorem for the $\cos(\bullet)$ function, namely

$$\cos(\alpha \pm \beta) = \cos(\alpha) \cos(\beta) \mp \sin(\alpha) \sin(\beta),$$

which gives the following for the choice $\alpha = \theta - \delta$, $\beta = \psi = \arctan(a/d_{so})$:

$$\begin{aligned} \mathbf{r} \cdot \mathbf{n}_\xi - \xi' &= r \cos \left(\overbrace{\theta - \delta}^{=\alpha} + \overbrace{\psi}^{=\beta} \right) - a \frac{d_{so}}{\sqrt{a^2 + d_{so}^2}} \\ &= r \cos(\theta - \delta) \cos(\psi) - r \sin(\theta - \delta) \sin(\psi) - a \frac{d_{so}}{\sqrt{a^2 + d_{so}^2}} \\ &= r \cos(\theta - \delta) \frac{d_{so}}{\sqrt{a^2 + d_{so}^2}} - (d_{so} + r \sin(\theta - \delta)) \frac{a}{\sqrt{a^2 + d_{so}^2}} \end{aligned} \quad (4.47)$$

with $\cos(\psi) \stackrel{(4.39)}{=} \frac{d_{so}}{\sqrt{a^2 + d_{so}^2}}$ and $\sin(\psi) \stackrel{\text{Fig.4.7}}{=} \frac{a}{\sqrt{a^2 + d_{so}^2}}.$

From the right-angled triangle in Figure 4.7 having $R = r \cos(\theta - \delta)$ as adjacent side and r as hypotenuse, we can follow the abbreviation

$$U := d_{so} + r \sin(\theta - \delta). \quad (4.48)$$

Now let's assume we have any selected position a' which is not necessarily identical to the recent position a indicating the X-ray of interest. Then the intercept theorem declares that

$$\tan(\psi') = \frac{a'}{d_{so}} = \frac{R'}{U} = \frac{r \cos(\theta - \delta)}{d_{so} + r \sin(\theta - \delta)} \Rightarrow U a' = r \cos(\theta - \delta) d_{so}. \quad (4.49)$$

If we now substitute equations (4.48) and (4.49) into equation (4.47), we obtain

$$\mathbf{r} \cdot \mathbf{n}_\xi - \xi' = (a' - a) \frac{U}{\sqrt{a^2 + d_{so}^2}}. \quad (4.50)$$

After we inserted the whole gained simplifications into equation (4.46), the reduced formulation reads

$$f(r, \delta) = \frac{1}{2} \int_0^{2\pi} \left\{ \int_{-a_{min}}^{a_{max}} \phi_\theta(a) g\left((a' - a) \frac{U}{\sqrt{a^2 + d_{so}^2}}\right) \left(\frac{d_{so}}{\sqrt{a^2 + d_{so}^2}}\right)^3 da \right\} d\theta, \quad (4.51)$$

in which the bracketed share constitutes a convolution with respect to the a -variable. The penultimate step towards the final form of the filtered backprojection reconstruction for fan beam scanning with a flat detector is to focus on the filter kernel $g(\bullet)$ in the spatial domain once again. Since

$$\begin{aligned} g(\xi) &= \int_{-\infty}^{\infty} |q| e^{i2\pi q \xi} dq \\ \Rightarrow g\left((a' - a) \frac{U}{\sqrt{a^2 + d_{so}^2}}\right) &= \int_{-\infty}^{\infty} |q| e^{i2\pi q (a' - a) \frac{U}{\sqrt{a^2 + d_{so}^2}}} dq, \end{aligned}$$

a coordinate substitution of the shape

$$q' = \frac{U}{\sqrt{a^2 + d_{so}^2}} q \Leftrightarrow q = \frac{\sqrt{a^2 + d_{so}^2}}{U} q' \quad (4.52)$$

yields the very helpful correlation

$$g\left((a' - a) \frac{U}{\sqrt{a^2 + d_{so}^2}}\right) = \frac{a^2 + d_{so}^2}{U^2} \int_{-\infty}^{\infty} |q'| e^{i2\pi q' (a' - a)} dq' = \frac{a^2 + d_{so}^2}{U^2} g(a' - a). \quad (4.53)$$

Condensing equations (4.51) and (4.53) gives

$$\begin{aligned} f(r, \delta) &= \frac{1}{2} \int_0^{2\pi} \frac{1}{U^2} \left\{ \int_{-a_{min}}^{a_{max}} \phi_\theta(a) g(a' - a) \frac{d_{so}^3}{\sqrt{a^2 + d_{so}^2}} da \right\} d\theta \\ &= \int_0^{2\pi} \frac{d_{so}^2}{U^2} \left\{ \underbrace{\frac{1}{2} \left(\phi_\theta(a) \frac{d_{so}}{\sqrt{a^2 + d_{so}^2}} \right) * g(a)}_{=: h_\theta(a)} \right\} d\theta. \end{aligned} \quad (4.54)$$

The underbraced convolution function $h_\theta(\bullet)$ represents the ramp filtering of the modified fan beam scanning projection data $\phi_\theta(\bullet)$ along the latest detector orientation (the \mathbf{e}_a -axis). Finally, the short version of the filtered backprojection can be specified as

$$f(r, \delta) = \int_0^{2\pi} \frac{d_{so}^2}{U^2} h_\theta(a) d\theta \quad \text{with} \quad h_\theta(a) = \frac{1}{2} \left(\phi_\theta(a) \frac{d_{so}}{\sqrt{a^2 + d_{so}^2}} \right) * g(a). \quad (4.55)$$

As already done at the end of subsection 4.4.2, we will also conclude this subsection with the key steps that have to be absolved to come from a set of available fan beam projection data $\phi_\theta(a)$ measured by a flat detector panel to local attenuation values $f(r, \delta)$ with the newly derived filtered backprojection reconstruction method.

- (1) Perform the coordinate transform from the ramp filter for a fan beam scenario to the ramp filter for parallel beam scanning, using relation (4.53).
- (2) Filter the projection data signal in the spatial domain along the \mathbf{e}_a -axis of the flat linear detector panel, giving the convolution function h_θ stated in equation (4.55).
- (3) Calculate the backprojection f by integration over all possible projection angles θ , using the essential variable U (length of the projection of the vector $\mathbf{r} - \mathbf{s}$ onto the current central ray of the fan beam): $f(r, \delta) = \int_0^{2\pi} \frac{d_{so}^2}{U^2} h_\theta(a) d\theta$.

4.4. 3D reconstruction methods for cone beam scanning

Basically, three-dimensional reconstruction needs a comprehensive theoretical derivation (please refer to the book Buzug, 2008 once again for more information) which is much longer than the already presented section 4.1 for two-dimensional reconstruction. Fortunately, the FDK cone beam reconstruction method (named after Feldkamp, Davis and Kress; Feldkamp, Davis, and Kress, 1984) can be introduced on top of the filtered backprojection reconstruction for a fan beam scanning scenario with a flat detector array given in the previous section 4.3. One could say the FDK reconstruction method is the direct generalization of the filtered backprojection for fan beam scanning onto cone beam projection data. Still, one important difference between the two reconstruction techniques is that the FDK reconstruction method can only give an approximation of the theoretically exactly solvable reconstruction problem. The reason for this is that the mathematically principally arbitrary path of the X-ray source $\mathbf{s}(\lambda)$ (which could be parametrized by a scalar path variable λ) is restricted to a circular motion around the fixed Cartesian $(\mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z)$ coordinate system in reconstruction with the FDK algorithm for practical purposes. But since the arising difference in the final reconstructed images is negligible for a general user and further focussing on the principles of three-dimensional reconstruction would lead much too far, we are content with the FDK reconstruction method in the following.

The reconstruction method proposed by Feldkamp, Davis and Kress is a derivative-free method which also works with a virtual flat detector (which is an extended two-dimensional rectangle and not only a one-dimensional array of detector elements this time) positioned in the iso-center of rotation \mathcal{O} . Of course the method could also be characterized for the curved variant of the detector in 3D, but this is not of interest at this point. The radiation source \mathbf{s} moves on a circle lying in the $(\mathbf{e}_x, \mathbf{e}_y)$ -plane with the radius d_{so} to the origin, so it still holds that $\mathbf{s} = (-d_{so} \sin(\theta), d_{so} \cos(\theta), 0)^T$. The projection angle θ is thereby determined by the angle between the central beam of the cone and the \mathbf{e}_y -axis. Furthermore, the measured projection values on the virtual detector are denoted by $\phi_\theta(a, b)$. The variables a, b define the position of a point $\mathbf{p} = (a, b)^T$ on the virtual flat detector panel along the principal $(\mathbf{e}_a, \mathbf{e}_b)$ -axes, starting in the center of the rectangle area of the detector,

for known angles ψ, κ_a , compare Figure 4.10(c) or 4.10(d).

The next step to approach the FDK reconstruction method lies in the introduction of the fan-specific coordinate system spanned by the normal vectors $\{\mathbf{n}_\sigma, \mathbf{n}_\eta, \mathbf{n}_\xi\}$, as depicted in Figure 4.9. The normal vector \mathbf{n}_σ points in direction of the \mathbf{e}_a -axis and is therefore rotated by the projection angle θ from the \mathbf{e}_x -axis in the \mathbf{e}_x - \mathbf{e}_y -plane. This rotational operation can be described mathematically by the orthogonal rotation matrix Σ . The normal vector \mathbf{n}_η simply points back to the X-ray source position along the central beam of the fan. Finally, the normal vector $\mathbf{n}_\xi = \mathbf{n}_\sigma \times \mathbf{n}_\eta$ is exactly the surface normal vector of the fan area \mathbf{A} . As we can see from Figure 4.9, this normal vector direction can also be generated by a rotation with the central cone beam angle $\kappa_0 = \arctan(b/d_{so})$ (see Figure 4.10(a)) in the \mathbf{e}_y - \mathbf{e}_z -plane starting from the \mathbf{e}_z -axis. Again, a formulation via an orthogonal rotation matrix Ξ can be given.

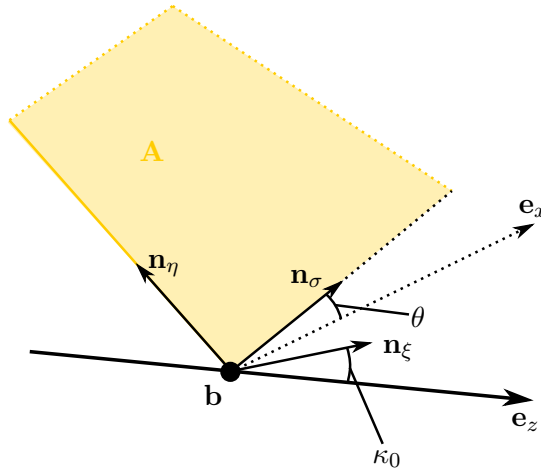


Figure 4.9.: illustration of the orientation of the new fan-specific coordinate system $(\mathbf{e}_\sigma, \mathbf{e}_\eta, \mathbf{e}_\xi)$ (which is still positioned in the global origin \mathcal{O}) by its spanning normal vectors $\{\mathbf{n}_\sigma, \mathbf{n}_\eta, \mathbf{n}_\xi\}$

Plugging all these preparations together, the coordinate changes between a point $\mathbf{r} = (x, y, z)^T$ in the fixed coordinate system $(\mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z)$ and its equivalent point $\tilde{\rho} = \mathbf{r}$ with coordinates $\tilde{\rho} = (r, s, t)^T$ on a specific fan beam surface \mathbf{A} in the $(\mathbf{e}_\sigma, \mathbf{e}_\eta, \mathbf{e}_\xi)$ coordinate system can be achieved using the following expressions:

$$\begin{aligned} \tilde{\rho} = (r, s, t)^T &= \Xi (\Sigma \mathbf{r} + \mathbf{b}) = \Xi \left(\Sigma (x, y, z)^T + (0, 0, b)^T \right) \\ \begin{bmatrix} r \\ s \\ t \end{bmatrix} &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\kappa_0) & \sin(\kappa_0) \\ 0 & -\sin(\kappa_0) & \cos(\kappa_0) \end{bmatrix} \left(\begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ b \end{bmatrix} \right) \\ &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\kappa_0) & \sin(\kappa_0) \\ 0 & -\sin(\kappa_0) & \cos(\kappa_0) \end{bmatrix} \begin{bmatrix} x \cos(\theta) + y \sin(\theta) \\ -x \sin(\theta) + y \cos(\theta) \\ z + b \end{bmatrix} \end{aligned}$$

$$\begin{aligned}
&= \begin{bmatrix} x \cos(\theta) + y \sin(\theta) \\ -x \sin(\theta) \cos(\kappa_0) + y \cos(\theta) \cos(\kappa_0) + (z+b) \sin(\kappa_0) \\ x \sin(\theta) \sin(\kappa_0) - y \cos(\theta) \sin(\kappa_0) + (z+b) \cos(\kappa_0) \end{bmatrix} \quad (4.56) \\
&\Leftrightarrow \mathbf{r} = (x, y, z)^T = \mathbf{\Sigma}^T (\mathbf{\Xi}^T \tilde{\rho} - \mathbf{b}) = \mathbf{\Sigma}^T \left(\mathbf{\Xi}^T (r, s, t)^T - (0, 0, b)^T \right) \\
&= \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \left(\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\kappa_0) & -\sin(\kappa_0) \\ 0 & \sin(\kappa_0) & \cos(\kappa_0) \end{bmatrix} \begin{bmatrix} r \\ s \\ t \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ b \end{bmatrix} \right) \\
&= \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r \\ s \cos(\kappa_0) - t \sin(\kappa_0) \\ s \sin(\kappa_0) + t \cos(\kappa_0) - b \end{bmatrix} \\
&= \begin{bmatrix} r \cos(\theta) - \sin(\theta) (s \cos(\kappa_0) - t \sin(\kappa_0)) \\ r \sin(\theta) + \cos(\theta) (s \cos(\kappa_0) - t \sin(\kappa_0)) \\ s \sin(\kappa_0) + t \cos(\kappa_0) - b \end{bmatrix} \quad (4.57)
\end{aligned}$$

If we focus on a selected fan \mathbf{A} and consider only some points $\mathbf{r} \in \mathbf{A}$, it holds that $t = 0$ along the \mathbf{e}_ξ -direction. Therewith, equation (4.57) can be simplified for points $\rho = (r, s)^T$ in this specific fan beam plane \mathbf{A} . We also use relations from Figure 4.10(a) to formulate equation (4.58):

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} r \cos(\theta) - s \sin(\theta) \cos(\kappa_0) \\ r \sin(\theta) + s \cos(\theta) \cos(\kappa_0) \\ s \sin(\kappa_0) - b \end{bmatrix} = \begin{bmatrix} r \cos(\theta) - s \sin(\theta) \frac{d_{so}}{\sqrt{d_{so}^2 + b^2}} \\ r \sin(\theta) + s \cos(\theta) \frac{d_{so}}{\sqrt{d_{so}^2 + b^2}} \\ s \frac{b}{\sqrt{d_{so}^2 + b^2}} - b \end{bmatrix}. \quad (4.58)$$

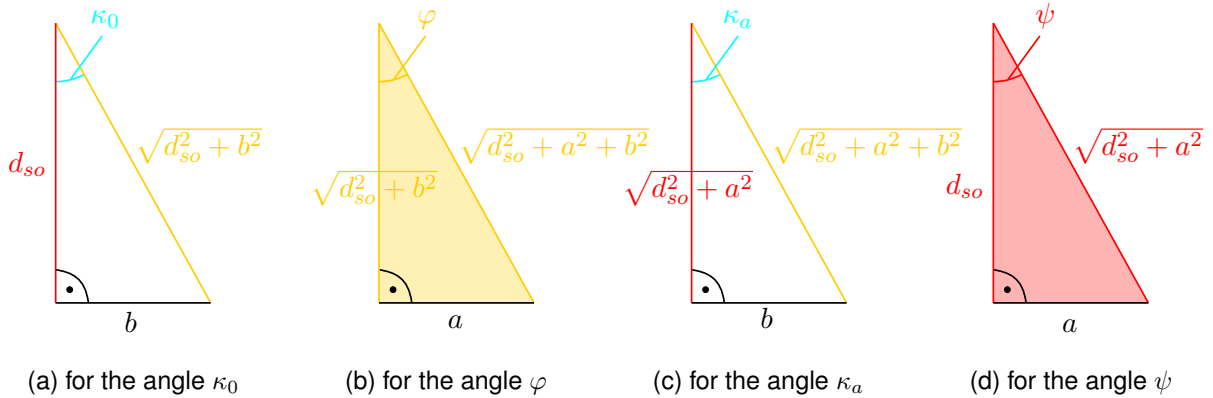


Figure 4.10.: focussing several right-angled triangles from Figure 4.8

For most of the subsequent steps of the FDK reconstruction algorithm, we can now use the almost perfect analogy to the filtered backprojection reconstruction method from section 4.3 on a fixed fan beam surface \mathbf{A} , for now parametrized by point coordinates $\rho = (r, s)^T$. We can directly exchange some of the used variables in the given order:

$$\{\xi, \psi, d_{so}, \sqrt{d_{so}^2 + a^2}, \theta\} \mapsto \{\zeta, \varphi, \sqrt{d_{so}^2 + b^2}, \sqrt{d_{so}^2 + a^2 + b^2}, \theta_b\}.$$

We indeed withdraw to the single fixed plane \mathbf{A} including the projection angle θ_b with the \mathbf{e}_y -axis (the angle index \bullet_b specifies that this selected plane intersects the \mathbf{e}_z -axis at $z = b$). Again, we are interested in the transfer from some kind of fan beam projection data $\phi_{\theta_b}(a, b)$ to the corresponding parallel geometry data $p_\gamma(\zeta)$ along the auxiliary \mathbf{e}_ζ -axis (recap Figure 4.8). For a deeper understanding of the specified relations, it is helpful to look at Figure 4.7 once again and to replace some identifiers as introduced above. The first analogous sizes are defined by

$$(4.39) \mapsto \zeta = a \cos(\varphi) = a \frac{\sqrt{d_{so}^2 + b^2}}{\sqrt{d_{so}^2 + a^2 + b^2}} \quad (4.59)$$

$$(4.40) \mapsto \gamma = \theta_b + \varphi = \theta_b + \arctan\left(a/\sqrt{d_{so}^2 + b^2}\right) \quad (4.60)$$

$$(4.41) \mapsto \rho \cdot \mathbf{n}_\zeta = \zeta = \rho \cos(\gamma - \delta). \quad (4.61)$$

To get nearer to the reconstruction algorithm, we transmit equation (4.44) to its new form

$$f(r, s) = \frac{1}{2} \int_0^{2\pi} \left(\int_{-\infty}^{\infty} p_\gamma(\zeta) g(\rho \cdot \mathbf{n}_\zeta - \zeta) d\zeta \right) d\gamma \quad (4.62)$$

with the addressed coordinate change to the auxiliary ζ -coordinate. Again, we will carry out the conversion from the $(\mathbf{e}_\zeta, \mathbf{e}_\gamma)$ to the detector-inherent $(\mathbf{e}_a, \mathbf{e}_{\theta_b})$ coordinate system by calculation of the Jacobian $d\zeta d\gamma \mapsto J da d\theta_b$. This yields

$$J = \begin{vmatrix} \frac{\partial \zeta}{\partial a} & \frac{\partial \zeta}{\partial \theta_b} \\ \frac{\partial \gamma}{\partial a} & \frac{\partial \gamma}{\partial \theta_b} \end{vmatrix} = \begin{vmatrix} \left(\frac{\sqrt{d_{so}^2 + b^2}}{\sqrt{d_{so}^2 + a^2 + b^2}} \right)^3 & 0 \\ \frac{\sqrt{d_{so}^2 + b^2}}{d_{so}^2 + a^2 + b^2} & 1 \end{vmatrix} = \left(\frac{\sqrt{d_{so}^2 + b^2}}{\sqrt{d_{so}^2 + a^2 + b^2}} \right)^3 = \cos(\varphi)^3, \quad (4.63)$$

following equation (4.45) and the auxiliary calculations below. Resubstitution of the achieved results in equation (4.62) gives the equation (4.64) similar to equation (4.46):

$$f(r, s) = \frac{1}{2} \int_0^{2\pi} \left\{ \int_{-a_{min}}^{a_{max}} \overbrace{p_{\theta_b + \varphi}}^{=\phi_{\theta_b}(a, b)|_{b=\text{const.}}} \underbrace{\left(a \frac{\sqrt{d_{so}^2 + b^2}}{\sqrt{d_{so}^2 + a^2 + b^2}} \right)}_{=\zeta} \cdots \right. \quad (4.64)$$

$$\left. g \left(\overbrace{\rho \cos \left(\underbrace{\theta_b + \arctan \left(\frac{a}{\sqrt{d_{so}^2 + b^2}} \right)}_{=\gamma} - \delta \right)}^{=\rho \cdot \mathbf{n}_\zeta - \zeta} - \underbrace{a \frac{\sqrt{d_{so}^2 + b^2}}{\sqrt{d_{so}^2 + a^2 + b^2}}}_{=\zeta} \right) \underbrace{\left(\frac{\sqrt{d_{so}^2 + b^2}}{\sqrt{d_{so}^2 + a^2 + b^2}} \right)^3}_{=J} da \right\} d\theta_b.$$

Notice the modified identity $p_\gamma(\zeta) = p_{\theta_b + \varphi} \left(a \sqrt{d_{so}^2 + b^2} / \sqrt{d_{so}^2 + a^2 + b^2} \right) = \phi_{\theta_b}(a, b)|_{b=\text{const.}}$ for the present three-dimensional scenario, where $b = \text{const.}$ references the specific fan beam plane \mathbf{A} under the projection angle θ_b . The next step on the way to the FDK reconstruction algorithm

is to work out the convolution integral in braces in equation (4.64). For this, the ensuing bunch of correlations similar to equations (4.47) through (4.50) is necessary:

$$\begin{aligned} \rho \cdot \mathbf{n}_\zeta - \zeta &= \rho \cos(\theta_b - \delta + \varphi) - \zeta \stackrel{(4.47)}{=} \rho \cos(\theta_b - \delta) \cos(\varphi) - \rho \sin(\theta_b - \delta) \sin(\varphi) - \zeta \\ &= \rho \cos(\theta_b - \delta) \frac{\sqrt{d_{so}^2 + b^2}}{\sqrt{d_{so}^2 + a^2 + b^2}} - \left(\sqrt{d_{so}^2 + b^2} + \rho \sin(\theta_b - \delta) \right) \frac{a}{\sqrt{d_{so}^2 + a^2 + b^2}} \end{aligned} \quad (4.65)$$

$$U_b \stackrel{(4.48)}{:=} \sqrt{d_{so}^2 + b^2} + \rho \sin(\theta_b - \delta) \quad (4.66)$$

$$\Rightarrow \tan(\varphi') = \frac{a'}{\sqrt{d_{so}^2 + b^2}} = \frac{\rho \cos(\theta_b - \delta)}{U_b} \Rightarrow U_b a' \stackrel{(4.49)}{=} \rho \cos(\theta_b - \delta) \sqrt{d_{so}^2 + b^2} \quad (4.67)$$

$$\Rightarrow \rho \cdot \mathbf{n}_\zeta - \zeta \stackrel{(4.50)}{=} (a' - a) \frac{U_b}{\sqrt{d_{so}^2 + a^2 + b^2}}. \quad (4.68)$$

All these expressions can be used to simplify equation (4.64) to

$$\begin{aligned} f(r, s) &\stackrel{(4.51)}{=} \frac{1}{2} \int_0^{2\pi} \left\{ \int_{-a_{min}}^{a_{max}} \phi_{\theta_b}(a, b) g \left((a' - a) \frac{U_b}{\sqrt{d_{so}^2 + a^2 + b^2}} \right) \dots \right. \\ &\quad \left. \left(\frac{\sqrt{d_{so}^2 + b^2}}{\sqrt{d_{so}^2 + a^2 + b^2}} \right)^3 da \right\} d\theta_b. \end{aligned} \quad (4.69)$$

Considering the filter kernel $g(\bullet)$ in the spatial domain reveals

$$g \left((a' - a) \frac{U_b}{\sqrt{d_{so}^2 + a^2 + b^2}} \right) \stackrel{(4.53)}{=} \frac{d_{so}^2 + a^2 + b^2}{U_b^2} g(a' - a) \quad (4.70)$$

with the simple substitution

$$q' = \frac{U_b}{\sqrt{d_{so}^2 + a^2 + b^2}} q \stackrel{(4.52)}{\Leftrightarrow} q = \frac{\sqrt{d_{so}^2 + a^2 + b^2}}{U_b} q'. \quad (4.71)$$

The last step directly obeying the principles of the two-dimensional filtered backprojection algorithm is therefore given by

$$\begin{aligned} f(r, s) &= \frac{1}{2} \int_0^{2\pi} \frac{1}{U_b^2} \left\{ \int_{-a_{min}}^{a_{max}} \phi_{\theta_b}(a, b) g(a' - a) \frac{\left(\sqrt{d_{so}^2 + b^2} \right)^3}{\sqrt{d_{so}^2 + a^2 + b^2}} da \right\} d\theta_b \\ &= \int_0^{2\pi} \frac{d_{so}^2 + b^2}{U_b^2} \left\{ \frac{1}{2} \left(\phi_{\theta_b}(a, b) \frac{\sqrt{d_{so}^2 + b^2}}{\sqrt{d_{so}^2 + a^2 + b^2}} \right) * g(a) \right\} d\theta_b. \end{aligned} \quad (4.72)$$

The missing puzzle piece to the final version of the FDK reconstruction method is to replace the dependency of the individual fan beam angle θ_b by the actual rotation angle θ of the sampling unit. On this occasion, we will also get rid of the fan beam surface coordinates r, s . Figure 4.11(a) clarifies that for two different source positions $\mathbf{s}_1, \mathbf{s}_2$ it holds that $\|\mathbf{s}_1\|_2 = \|\mathbf{s}_2\|_2 = d_{so}$ and so $\|\mathbf{s}_1 - \mathbf{b}\|_2 = \|\mathbf{s}_2 - \mathbf{b}\|_2 = \sqrt{d_{so}^2 + b^2}$. We can follow that for small angular changes $\Delta\theta$

$$\|\mathbf{s}_1 - \mathbf{s}_2\|_2 = d_{so} \Delta\theta \approx \sqrt{d_{so}^2 + b^2} \Delta\theta_b \Rightarrow \Delta\theta_b \approx \frac{d_{so}}{\sqrt{d_{so}^2 + b^2}} \Delta\theta.$$

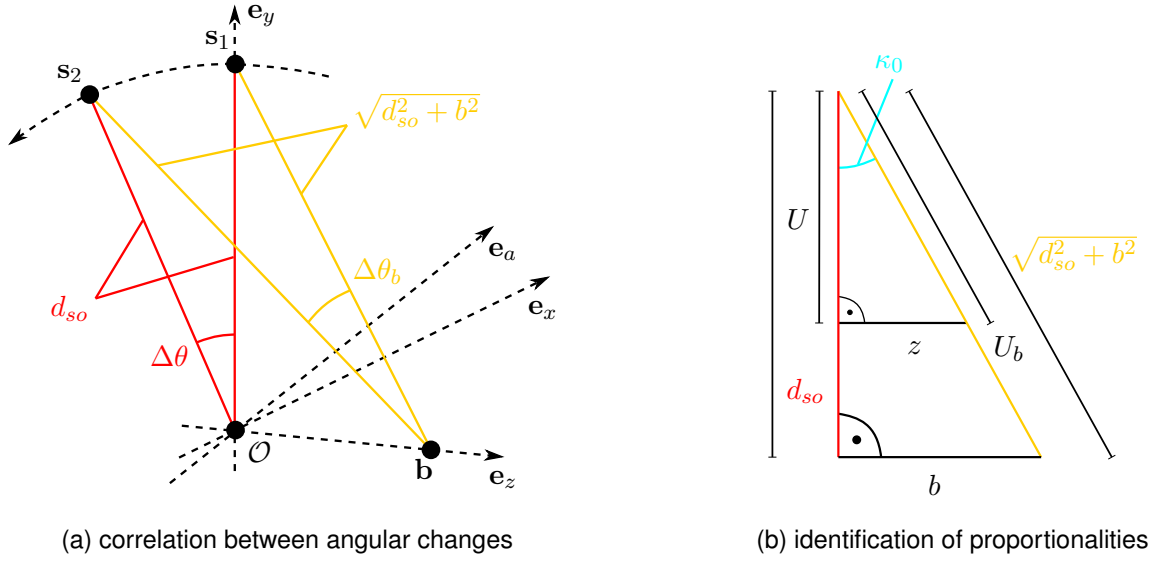


Figure 4.11.: additional illustration of geometrical relations used to derive the FDK algorithm

Passing over to an infinitesimal formulation, this relation in combination with equation (4.72) yields

$$d\theta_b = \frac{d_{so}}{\sqrt{d_{so}^2 + b^2}} d\theta \quad (4.73)$$

$$\Rightarrow f(r, s) = \int_0^{2\pi} \underbrace{\frac{d_{so}^2 + b^2}{U_b^2}}_{(*)} \left\{ \frac{1}{2} \left(\phi_\theta(a, b) \underbrace{\frac{d_{so}}{\sqrt{d_{so}^2 + a^2 + b^2}}}_{=: h_\theta(a, b)} * g(a) \right) \right\} d\theta. \quad (4.74)$$

The termed marked with (*) can be identified with the fracture d_{so}^2/U^2 using the intercept theorem for the triangle sketched in Figure 4.11(b). This identity reduces equation (4.74) to its short form

$$f(r, s) = \int_0^{2\pi} \frac{d_{so}^2}{U^2} h_\theta(a, b) d\theta. \quad (4.75)$$

The weighting factor $d_{so}/\sqrt{d_{so}^2 + a^2 + b^2}$ in the high-pass filtered function $h_\theta(\bullet)$ could also be expressed in terms of angles like $\cos(\beta)$ or $\cos(\psi) \cos(\kappa_a)$, which can be derived with a sharp look at Figure 4.8 or 4.10. Still, we didn't resolve the full three-dimensional formulation of the FDK reconstruction algorithm: We eventually have to sort each single fan beam surface \mathbf{A} into the fixed Cartesian $(\mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z)$ -coordinate system, especially the z -component of the reconstruction has not been treated yet. From equation (4.58) it holds that

$$z = s \frac{b}{\sqrt{d_{so}^2 + b^2}} - b,$$

which is quite badly invertible. A clever choice can be found with a look at Figure 4.11(b):

$$\tan(\kappa_0) = \frac{b}{d_{so}} = \frac{z}{U} \Leftrightarrow b = \frac{d_{so}}{U} z. \quad (4.76)$$

In the end, it is reasonable to write the FDK reconstruction method in the symbolic form

$$f(x, y, z) = \int_0^{2\pi} \frac{d_{so}^2}{U(x, y, \theta)^2} h_\theta(a(x, y, z, \theta), b(x, y, z, \theta)) d\theta. \quad (4.77)$$

The conclusion of the three-dimensional reconstruction algorithm by Feldkamp, Davis and Kress starting from a set of available cone beam projection data $\phi_\theta(a, b)$ measured with a flat rectangular detector panel makes the local attenuation values $f(x, y, z)$ accessible using the subsequent key steps.

- (1) Perform the coordinate transform from the ramp filter for a cone beam scenario to the ramp filter for parallel beam scanning, using relation (4.70).
- (2) Filter the projection data signal in the spatial domain along the e_a -axis of the flat rectangular detector, giving the weighted convolution function h_θ stated in equation (4.74).
- (3) Calculate the filtered backprojection f by integration over all possible projection angles θ , using the essential variable $U(\bullet)$ (length of the projection of the vector $\mathbf{r} - \mathbf{s}$ onto the current, non-angulated central ray of the cone beam):

$$f(x, y, z) = \int_0^{2\pi} \frac{d_{so}^2}{U(x, y, \theta)^2} h_\theta(a(x, y, z, \theta), b(x, y, z, \theta)) d\theta$$

$$\text{with } b(x, y, z, \theta) = \frac{d_{so}}{U(x, y, \theta)} z.$$

4.5. Algebraic reconstruction methods in 2D and 3D

A completely different approach to the reconstruction of the object's attenuation coefficients of interest starting with a given set of projection data obtained by measuring with a X-ray detector is the class of algebraic reconstruction methods. In the following, the question whether we are dealing with two- or three-dimensional data sets becomes insignificant because this facts just expresses itself in an increase of dimensionality. The key virtues of algebraic reconstruction in contrast to analytical reconstruction methods are their flexibility in geometrical formulation and ability to make use of some prior knowledge information that can be included in the solution process. The backside are the mostly higher computational efforts, depending on the iteration count, which of course competes the resulting image quality of the reconstruction.

The problem of algebraic reconstruction can be poured into the single equation

$$\mathbf{W}\mathbf{v} = \mathbf{p}, \quad (4.78)$$

wherein we have the vector $\mathbf{v} \in \mathbb{R}^n$ of voxel data, the vector $\mathbf{p} \in \mathbb{R}^m$ containing the acquired projection data and finally the projection matrix $\mathbf{W} \in \mathbb{R}^{m \times n}$. Each projection weight w_{ij} symbolizes the influence of the voxel v_j onto the specific measured X-ray beam p_i - therefore most of the matrix entries vanish (because only a few voxels contribute to the information of a sole ray), so \mathbf{W} is in general a sparse matrix. To determine such a contribution, there exist several so-called kernels which are not of any deeper importance here, so let us just assume they are given. Equation (4.78) constitutes an algebraic forward projection of some known voxel data onto specific projection data, which naturally depend on the defined projection weights inside \mathbf{W} . The opposed procedure starting from given projection information can again be denoted as a backprojection and constituted mathematically via

$$\mathbf{u} = \mathbf{W}^T \mathbf{p}. \quad (4.79)$$

Be aware that equation (4.79) does not recover the full voxel information, as it has already been the case for analytical reconstruction with the simple backprojection ansatz in subsection 4.2.1. However, a generic extension like the filtered backprojection in subsection 4.2.2 synthesized on top of the simple backprojection is not required here. The goal of algebraic reconstruction rather

is to find “ $\mathbf{v}^* = \mathbf{W}^{-1}\mathbf{p}$ ”, but since the projection matrix is not square and anyways too large to invert algebraically, “ \mathbf{W}^{-1} ” is not available. The solution of this challenging point is to set up the optimization problem

$$\text{find } \mathbf{v}^* \in \mathbb{R}^n \text{ so that } \mathbf{v}^* = \arg \min_{\mathbf{v} \in \mathbb{R}^n} \|\mathbf{p} - \mathbf{W}\mathbf{v}\|^2 \quad (4.80)$$

which searches for the voxel vector \mathbf{v}^* that minimizes the projection distance in a to-be-chosen vector norm $\|\bullet\|$. The broad field of Numerical linear algebra shows some satisfactory procedures to dissolve the imposed optimization problem of equation (4.80).

We will focus on the simultaneous iterative reconstruction technique (SIRT) from the family of Landweber algorithms and on the Krylov subspace algorithm conjugate gradients for least squares (CGLS) in the subsequent two subsections. This is of course motivated by the fact that these two algebraic reconstruction algorithms are available for all imaginable configurations of the ASTRA toolbox. Moreover, the circumstance that other possible algebraic reconstruction algorithms like the algebraic reconstruction technique (ART) or the simultaneous algebraic reconstruction technique (SART) (for a short characterization, see Kak and Slaney, 2001, p. 283 ff.) can only be used as algorithms for projection data from two-dimensional scanning setups suggests that those ones cannot reliably handle the increased requirements of the more demanding general three-dimensional reconstruction task. The two main disparities between SIRT and CGLS are the differing mathematical backgrounds from which the methods are evolved on the one hand and the conflict between occupied working memory and necessary iteration count to achieve reconstructions of adequate image quality on the other hand. While the SIRT algorithm is lightly more memory-efficient, CGLS reaches good reconstructions for a considerably lower number of iterations (and is therefore linked to a shorter computational time).

4.5.1. Simultaneous iterative reconstruction technique (SIRT)

We will unceremoniously introduce the iteration necessary to update the i -th estimation of the voxels' attenuation coefficient vector $\mathbf{v}^{(i)}$ to the $(i+1)$ -st step using the SIRT reconstruction algorithm, following the paper (Gregor and Benson, 2008):

$$\begin{aligned} v_j^{(i+1)} &= v_j^{(i)} + \frac{\sum_{k=1}^m \left[w_{kj} \left(p_k - \sum_{l=1}^n w_{kl} v_l^{(i)} \right) / \sum_{l=1}^n w_{kl} \right]}{\sum_{k=1}^m w_{kj}} \quad \text{for } j = 1, \dots, n \\ \Leftrightarrow \quad \mathbf{v}^{(i+1)} &= \mathbf{v}^{(i)} + \mathbf{C}\mathbf{W}^T \mathbf{R} (\mathbf{p} - \mathbf{W}\mathbf{v}^{(i)}) . \end{aligned} \quad (4.81)$$

Herein, the currently iterated projection distance $\mathbf{p} - \mathbf{W}\mathbf{v}^{(i)}$ is concatenated with a particular sequence of matrix multiplications, where the diagonal matrices \mathbf{C} and \mathbf{R} contain the inverse column sum $c_{jj} = 1 / \sum_{k=1}^m w_{kj}$ respectively the inverse row sum $r_{kk} = 1 / \sum_{l=1}^n w_{kl}$ of the projection matrix \mathbf{W} . The SIRT procedure given in equation (4.81) indeed solves an optimization problem comparable to the one in equation (4.80), namely

$$\mathbf{v}^* = \arg \min_{\mathbf{v} \in \mathbb{R}^n} \|\mathbf{p} - \mathbf{W}\mathbf{v}\|_R^2 \quad \text{with} \quad \|\mathbf{p} - \mathbf{W}\mathbf{v}\|_R^2 := (\mathbf{p} - \mathbf{W}\mathbf{v})^T \mathbf{R} (\mathbf{p} - \mathbf{W}\mathbf{v}) .$$

To analyze the convergence, we need to reshape the SIRT specification to normal equation form to access an eigenvalue consideration. This yields the set of relations

$$\text{normal equation: } \mathbf{W}^T \mathbf{R} \mathbf{W} \mathbf{v} = \mathbf{W}^T \mathbf{R} \mathbf{p} \xrightarrow{\mathbf{C}(\bullet)} \mathbf{C} \mathbf{W}^T \mathbf{R} \mathbf{W} \mathbf{v} = \mathbf{C} \mathbf{W}^T \mathbf{R} \mathbf{p} \quad (4.82)$$

$$\text{splitting: } \mathbf{C} \mathbf{W}^T \mathbf{R} \mathbf{W} = \mathbf{I} - (\mathbf{I} - \mathbf{C} \mathbf{W}^T \mathbf{R} \mathbf{W}) \quad (4.83)$$

$$\Rightarrow \mathbf{v}^{(i+1)} = (\mathbf{I} - \mathbf{C} \mathbf{W}^T \mathbf{R} \mathbf{W}) \mathbf{v}^{(i)} + \mathbf{C} \mathbf{W}^T \mathbf{R} \mathbf{p} \stackrel{(4.81)}{=} \mathbf{v}^{(i+1)} = \mathbf{v}^{(i)} + \mathbf{C} \mathbf{W}^T \mathbf{R} (\mathbf{p} - \mathbf{W} \mathbf{v}^{(i)}).$$

The splitting introduced in equation (4.83) is essential for the discussion of eigenvalue properties, and as shown above also reflects the SIRT algorithm. The convergence of the iterative SIRT method is guaranteed mathematically if

$$\varrho(\mathbf{I} - \mathbf{C} \mathbf{W}^T \mathbf{R} \mathbf{W}) = \max_{\lambda \in \Lambda} |1 - \lambda| \stackrel{!}{<} 1, \quad (4.84)$$

that is the largest eigenvalue of the iteration matrix has a strict upper bound of 1. On the one hand, the matrix $\mathbf{C} \mathbf{W}^T \mathbf{R} \mathbf{W}$ has strictly positive eigenvalues $\lambda > 0$ (\mathbf{C} is positive definite and if \mathbf{W} has full column rank, $\mathbf{W}^T \mathbf{R} \mathbf{W}$ is even symmetric positive definite). On the other hand, the equivalence of the spectral radius $\varrho(\bullet)$ with any vector norm on finite dimensional vector spaces induces

$$\varrho(\mathbf{C} \mathbf{W}^T \mathbf{R} \mathbf{W}) \leq \|\mathbf{C} \mathbf{W}^T\|_{\infty} \|\mathbf{R} \mathbf{W}\|_{\infty}$$

when choosing the row sum matrix norm $\|\bullet\|_{\infty}$. Thanks to the construction of the matrices \mathbf{C} and \mathbf{R} , both expressions equal 1. Altogether, the SIRT algorithm given in equation (4.81) globally converges when using an arbitrary starting point $\mathbf{v}^{(0)}$ because the two previous points show

$$0 \leq |1 - \lambda| < 1,$$

fulfilling the necessary condition for convergence in equation (4.84).

4.5.2. Conjugate gradients for least squares (CGLS)

The CGLS method comes from the class of Krylov subspace methods and therefore follows deviant mathematical principles. Just as before, we are interested in solving the least squares optimization problem (4.80), this time applying the Euclidean vector norm $\|\bullet\|_2$. Instead of equation (4.78), it is adequate to solve the corresponding normal equation

$$\mathbf{W}^T \mathbf{W} \mathbf{v} = \mathbf{W}^T \mathbf{p}. \quad (4.85)$$

To make the algorithmic steps of the CGLS method a bit clearer (and to pick up the train of thoughts from Kloek, 2012), the method of steepest decent for the least square (SDLS) problem will be specified first.

The SDLS method is only employable for symmetric positive definite matrices, so it is necessary to start with the normal equation (4.85) ($\mathbf{W}^T \mathbf{W}$ is of course symmetric positive definite). The quadratic form

$$f(\mathbf{v}) = \mathbf{v}^T (\mathbf{W}^T \mathbf{W}) \mathbf{v} - 2 (\mathbf{W}^T \mathbf{p})^T \mathbf{v} \quad (4.86)$$

is exploited to find the minimum of the least squares optimization problem. Because of its quadratic (parabolical) characteristic, the global minimum of $f(\bullet)$ can be obtained by evaluating

$$f'(\mathbf{v}) = 2 \mathbf{W}^T \mathbf{W} \mathbf{v} - 2 \mathbf{W}^T \mathbf{p} \stackrel{!}{=} \mathbf{0}, \quad (4.87)$$

and reshaping of this condition indite yields the normal equation (4.85). Thus, finding the minimum of the normal equation can equivalently be done as an iterative search for the zero point of $f'(\bullet)$. If we start with a given initial guess $\mathbf{v}^{(0)}$ and follow the direction of steepest decent $-f'(\mathbf{v}^{(i)}) = 2\mathbf{W}^T \mathbf{p} - 2\mathbf{W}^T \mathbf{W} \mathbf{v}^{(i)}$ in the iteration from $\mathbf{v}^{(i)}$ to $\mathbf{v}^{(i+1)}$, we can define the following two important sizes for the SDLS algorithm:

$$i\text{-th residual vector for (4.78): } \mathbf{r}^{(i)} := \mathbf{p} - \mathbf{W} \mathbf{v}^{(i)} \quad (4.88)$$

$$i\text{-th residual vector for (4.85): } \mathbf{s}^{(i)} := \mathbf{W}^T \mathbf{p} - \mathbf{W}^T \mathbf{W} \mathbf{v}^{(i)} = \mathbf{W}^T \mathbf{r}^{(i)} = -\frac{1}{2} f'(\mathbf{v}^{(i)}). \quad (4.89)$$

The iteration rule of SDLS takes the α_i -share of the step $\mathbf{s}^{(i)}$ to find the new voxel values $\mathbf{v}^{(i+1)} = \mathbf{v}^{(i)} + \alpha_i \mathbf{s}^{(i)}$. This proportion has to be ascertained by minimizing the function value $f(\mathbf{v}^{(i+1)})$ with respect to α_i via

$$\begin{aligned} 0 &\stackrel{!}{=} \frac{d}{d\alpha_i} f(\mathbf{v}^{(i+1)}) = f'(\mathbf{v}^{(i+1)})^T \frac{d}{d\alpha_i} \mathbf{v}^{(i+1)} \stackrel{(4.89)}{=} -2\mathbf{s}^{(i+1)T} \mathbf{s}^{(i)} \\ &\stackrel{(-\frac{1}{2})}{\Rightarrow} 0 \stackrel{!}{=} \mathbf{s}^{(i+1)T} \mathbf{s}^{(i)} \stackrel{(4.89)}{=} \left[\mathbf{W}^T \mathbf{p} - \mathbf{W}^T \mathbf{W} \underbrace{(\mathbf{v}^{(i)} + \alpha_i \mathbf{s}^{(i)})}_{\mathbf{v}^{(i+1)}} \right]^T \mathbf{s}^{(i)} \\ &= \left(\underbrace{\mathbf{W}^T \mathbf{p} - \mathbf{W}^T \mathbf{W} \mathbf{v}^{(i)}}_{\mathbf{s}^{(i)}} \right)^T \mathbf{s}^{(i)} - \alpha_i \mathbf{s}^{(i)T} \mathbf{W}^T \mathbf{W} \mathbf{s}^{(i)} \Rightarrow \alpha_i = \frac{\mathbf{s}^{(i)T} \mathbf{s}^{(i)}}{\mathbf{s}^{(i)T} \mathbf{W}^T \mathbf{W} \mathbf{s}^{(i)}}. \quad (4.90) \end{aligned}$$

To conclude this short excursion to the SDLS method, its algorithmic scheme is stated as follows:

$$\begin{aligned} \mathbf{r}^{(0)} &= \mathbf{p} - \mathbf{W} \mathbf{v}^{(0)} \rightarrow \mathbf{s}^{(0)} = \mathbf{W}^T \mathbf{r}^{(0)} \\ \alpha_i &= \frac{\mathbf{s}^{(i)T} \mathbf{s}^{(i)}}{\mathbf{s}^{(i)T} \mathbf{W}^T \mathbf{W} \mathbf{s}^{(i)}} \rightarrow \mathbf{v}^{(i+1)} = \mathbf{v}^{(i)} + \alpha_i \mathbf{s}^{(i)} \rightarrow \mathbf{r}^{(i+1)} = \mathbf{r}^{(i)} - \alpha_i \mathbf{W} \mathbf{s}^{(i)} \rightarrow \mathbf{s}^{(i+1)} = \mathbf{W}^T \mathbf{r}^{(i+1)}. \end{aligned}$$

The expression for $\mathbf{r}^{(i+1)}$ herein can be derived by

$$\mathbf{r}^{(i+1)} = \mathbf{p} - \mathbf{W} \underbrace{(\mathbf{v}^{(i)} + \alpha_i \mathbf{s}^{(i)})}_{\mathbf{v}^{(i+1)}} = \underbrace{\mathbf{p} - \mathbf{W} \mathbf{v}^{(i)}}_{\mathbf{r}^{(i)}} - \alpha_i \mathbf{W} \mathbf{s}^{(i)}.$$

Important to keep in mind at this point is that the factor α_i in equation (4.90) minimizes the norm of the residual vector $\|\mathbf{r}^{(i+1)}\|_2^2 = \|\mathbf{r}^{(i)} - \alpha_i \mathbf{W} \mathbf{s}^{(i)}\|_2^2$ with knowledge of the previous iteration.

The conjugate gradients for least squares (CGLS) method exhibits a similar property, as we will derive in its entirety in this subsection. After we developed the algorithmic steps for SDLS in a “forward” way, for CGLS we will proceed inversely by instantaneously declaring the algorithmic scheme:

$$\begin{aligned} \mathbf{r}^{(0)} &= \mathbf{p} - \mathbf{W} \mathbf{v}^{(0)} \rightarrow \mathbf{s}^{(0)} = \mathbf{W}^T \mathbf{r}^{(0)} =: \mathbf{t}^{(0)} \\ \mathbf{v}^{(i+1)} &= \mathbf{v}^{(i)} + \alpha_i \mathbf{t}^{(i)} \rightarrow \mathbf{r}^{(i+1)} = \mathbf{r}^{(i)} - \alpha_i \mathbf{W} \mathbf{t}^{(i)} \rightarrow \mathbf{s}^{(i+1)} = \mathbf{W}^T \mathbf{r}^{(i+1)} \rightarrow \mathbf{t}^{(i+1)} = \mathbf{s}^{(i+1)} + \beta_i \mathbf{t}^{(i)}. \end{aligned}$$

The equal signs shall be understood as definitions by now, and we single out the relation

$$\mathbf{s}^{(i+1)} = \mathbf{W}^T \mathbf{r}^{(i+1)} = \mathbf{s}^{(i)} - \alpha_i \mathbf{W}^T \mathbf{W} \mathbf{t}^{(i)}. \quad (4.91)$$

The factor α_i of the algorithm does not match the one worked out for SDLS and the new factor β_i is yet unknown, so we have to impose the first condition that $\|\mathbf{r}^{(i+1)}\|_2^2$ should be minimized with respect to the factors α_i **and** α_{i-1} . Since

$$\mathbf{r}^{(i+1)} = \mathbf{r}^{(i)} - \alpha_i \mathbf{W} \mathbf{t}^{(i)} = \mathbf{r}^{(i-1)} - \alpha_{i-1} \mathbf{W} \mathbf{t}^{(i-1)} - \alpha_i \mathbf{W} \mathbf{t}^{(i)}, \quad (4.92)$$

the norm of the subsequent residual vector becomes

$$\begin{aligned} \|\mathbf{r}^{(i+1)}\|_2^2 &= \mathbf{r}^{(i+1)T} \mathbf{r}^{(i+1)} \stackrel{(4.92)}{=} \mathbf{r}^{(i-1)T} \mathbf{r}^{(i-1)} - 2\alpha_{i-1} \mathbf{r}^{(i-1)T} \mathbf{W} \mathbf{t}^{(i-1)} - 2\alpha_i \mathbf{r}^{(i-1)T} \mathbf{W} \mathbf{t}^{(i)} \\ &\quad + \alpha_{i-1}^2 \mathbf{t}^{(i-1)T} \mathbf{W}^T \mathbf{W} \mathbf{t}^{(i-1)} + \cancel{2\alpha_{i-1}\alpha_i \mathbf{t}^{(i-1)T} \mathbf{W}^T \mathbf{W} \mathbf{t}^{(i)}} + \alpha_i^2 \mathbf{t}^{(i)T} \mathbf{W}^T \mathbf{W} \mathbf{t}^{(i)}. \end{aligned} \quad (4.93)$$

In order to regard the minimization problem independently for α_{i-1} or α_i , we omit the canceled term which contains the mixed product $\alpha_{i-1}\alpha_i$ in equation (4.93). To make this possible, a second additional condition has to be fixed: we have to choose the factor β_i in the relation $\mathbf{t}^{(i+1)} = \mathbf{s}^{(i+1)} + \beta_i \mathbf{t}^{(i)}$ in such a way that

$$\mathbf{t}^{(i+1)T} \mathbf{W}^T \mathbf{W} \mathbf{t}^{(i)} \stackrel{!}{=} 0. \quad (4.94)$$

After this requirement has been recorded, we can come back to the investigation of equation (4.93), which is now differentiated with respect to α_i (it is not necessary to consider α_{i-1} because this should have been treated in the previous iteration):

$$\begin{aligned} 0 &\stackrel{!}{=} \frac{d}{d\alpha_i} \|\mathbf{r}^{(i+1)}\|_2^2 = -2\mathbf{r}^{(i-1)T} \mathbf{W} \mathbf{t}^{(i)} + 2\alpha_i \mathbf{t}^{(i)T} \mathbf{W}^T \mathbf{W} \mathbf{t}^{(i)} \\ \Rightarrow \alpha_i &= \frac{\mathbf{r}^{(i-1)T} \mathbf{W} \mathbf{t}^{(i)}}{\mathbf{t}^{(i)T} \mathbf{W}^T \mathbf{W} \mathbf{t}^{(i)}} \stackrel{(4.91)}{=} \frac{\mathbf{s}^{(i-1)T} \mathbf{t}^{(i)}}{\tau^{(i)}} \stackrel{(4.91)}{=} \frac{\mathbf{s}^{(i)T} \mathbf{t}^{(i)}}{\tau^{(i)}} + \alpha_{i-1} \frac{\mathbf{t}^{(i-1)T} \mathbf{W}^T \mathbf{W} \mathbf{t}^{(i)}}{\tau^{(i)}}. \end{aligned} \quad (4.95)$$

The size $\tau^{(i)}$ is just an abbreviation for the denominator, and equation (4.95) is not the final shape of the factor α_i . Step by step, a small set of lemmata will be proved in the following, which helps us to formulate the CGLS method further.

The first claim comprises $\mathbf{t}^{(i)} \perp \mathbf{s}^{(i+1)}$, which can be shown quite easily:

$$\mathbf{t}^{(i)T} \mathbf{s}^{(i+1)} \stackrel{(4.91)}{=} \mathbf{t}^{(i)T} \mathbf{s}^{(i)} - \underbrace{\frac{\mathbf{s}^{(i)T} \mathbf{t}^{(i)}}{\tau^{(i)}} \mathbf{t}^{(i)T} \mathbf{W}^T \mathbf{W} \mathbf{t}^{(i)}}_{\alpha_i} = \mathbf{t}^{(i)T} \mathbf{s}^{(i)} - \mathbf{s}^{(i)T} \mathbf{t}^{(i)} = 0. \quad (4.96)$$

This helps us proving the deduction $\mathbf{s}^{(i)T} \mathbf{t}^{(i)} = \mathbf{s}^{(i)T} \mathbf{s}^{(i)}$, which will be done for the $(i+1)$ -st iteration for readability reasons.

$$\mathbf{s}^{(i+1)T} \mathbf{t}^{(i+1)} \stackrel{(4.96)}{=} \mathbf{s}^{(i+1)T} \mathbf{s}^{(i+1)} + \beta_i \mathbf{s}^{(i+1)T} \mathbf{t}^{(i)} \xrightarrow{(i+1) \rightarrow i} \mathbf{s}^{(i)T} \mathbf{t}^{(i)} = \mathbf{s}^{(i)T} \mathbf{s}^{(i)} \quad (4.97)$$

With this understanding, the factor α_i can be transformed to its final form

$$\alpha_i \stackrel{(4.95)}{=} \frac{\mathbf{s}^{(i)T} \mathbf{t}^{(i)}}{\tau^{(i)}} \stackrel{(4.97)}{=} \frac{\mathbf{s}^{(i)T} \mathbf{s}^{(i)}}{\tau^{(i)}}. \quad (4.98)$$

Another intermediate lemma claims the one-liner

$$\begin{aligned} \tau^{(i+1)} &= \mathbf{t}^{(i+1)T} \mathbf{W}^T \mathbf{W} \mathbf{t}^{(i+1)} \stackrel{(4.94)}{=} \mathbf{t}^{(i+1)T} \mathbf{W}^T \mathbf{W} \mathbf{s}^{(i+1)} + \beta_i \mathbf{t}^{(i+1)T} \mathbf{W}^T \mathbf{W} \mathbf{t}^{(i)} = \mathbf{t}^{(i+1)T} \mathbf{W}^T \mathbf{W} \mathbf{s}^{(i+1)} \\ &\xrightarrow{(i+1) \rightarrow i} \mathbf{t}^{(i)T} \mathbf{W}^T \mathbf{W} \mathbf{s}^{(i)} = \mathbf{t}^{(i)T} \mathbf{W}^T \mathbf{W} \mathbf{t}^{(i)} = \tau^{(i)}, \end{aligned} \quad (4.99)$$

where the second imposed condition was the central ingredient again and of course $(i+1)$ instead of i was used for consistency. More meaningful for the CGLS routine is the conclusion that $\mathbf{s}^{(i)} \perp \mathbf{s}^{(i+1)}$, for which the prove is still uncomplicated:

$$\mathbf{s}^{(i)T} \mathbf{s}^{(i+1)} \stackrel{(4.91)}{=} \mathbf{s}^{(i)T} \mathbf{s}^{(i)} - \alpha_i \mathbf{s}^{(i)T} \mathbf{W}^T \mathbf{W} \mathbf{t}^{(i)} \stackrel{(4.98)}{=} \mathbf{s}^{(i)T} \mathbf{s}^{(i)} - \frac{\mathbf{s}^{(i)T} \mathbf{s}^{(i)}}{\tau^{(i)}} \tau^{(i)} = 0. \quad (4.100)$$

Finally, all lemmata that are necessary to study equation (4.94) and especially the β_i -factor are prepared. We require

$$0 \stackrel{!}{=} \mathbf{t}^{(i)T} \mathbf{W}^T \mathbf{W} \mathbf{t}^{(i+1)} = \mathbf{t}^{(i)T} \mathbf{W}^T \mathbf{W} \mathbf{s}^{(i+1)} + \beta_i \underbrace{\mathbf{t}^{(i)T} \mathbf{W}^T \mathbf{W} \mathbf{t}^{(i)}}_{=\tau^{(i)}} \Rightarrow \beta_i = \frac{-\mathbf{t}^{(i)T} \mathbf{W}^T \mathbf{W} \mathbf{s}^{(i+1)}}{\tau^{(i)}}$$

and use the subsequent side reflection of the iteration instruction

$$\mathbf{W} \mathbf{t}^{(i)} \stackrel{(4.92)}{=} \frac{\mathbf{r}^{(i)} - \mathbf{r}^{(i+1)}}{\alpha_i} \Rightarrow -\mathbf{t}^{(i)T} \mathbf{W}^T \mathbf{W} = \frac{(\mathbf{r}^{(i+1)} - \mathbf{r}^{(i)})^T \mathbf{W}}{\alpha_i} \stackrel{(4.91)}{=} \frac{(\mathbf{s}^{(i+1)} - \mathbf{s}^{(i)})^T}{\alpha_i}$$

to obtain the factor β_i via

$$\beta_i \stackrel{(4.98)}{=} \stackrel{(4.100)}{\left(\frac{\mathbf{s}^{(i+1)T} \mathbf{s}^{(i+1)}}{\cancel{\tau^{(i)}}} - \frac{\mathbf{s}^{(i)T} \mathbf{s}^{(i+1)}}{\cancel{\tau^{(i)}}} \right) \frac{\cancel{\tau^{(i)}}}{\mathbf{s}^{(i)T} \mathbf{s}^{(i)}}} = \frac{\mathbf{s}^{(i+1)T} \mathbf{s}^{(i+1)}}{\mathbf{s}^{(i)T} \mathbf{s}^{(i)}}. \quad (4.101)$$

If the two factors α_i from equation (4.98) and β_i from (4.101) are inserted into the algorithmic scheme with which we started the previous derivation, the iterative CGLS method is completed. It remains to explain why CGLS belongs to the family of Krylov subspace methods: By its construction in the additional condition (4.94), it holds that $\mathbf{W} \mathbf{t}^{(i+1)} \perp \mathbf{W} \mathbf{t}^{(i)}$ and from means of iteration also $\mathbf{W} \mathbf{t}^{(i+1)} \perp \{\mathbf{W} \mathbf{t}^{(i)}, \dots, \mathbf{W} \mathbf{t}^{(0)}\}$. This means the vector $\mathbf{W} \mathbf{t}^{(i+1)}$ is orthogonal to the subspace $\text{span}\{\mathbf{W} \mathbf{t}^{(i)}, \dots, \mathbf{W} \mathbf{t}^{(0)}\}$, which exactly determines the characteristic of the powerful Krylov subspace methods. Moreover, with regard to the least squares problem formulation, the residual $\|\mathbf{r}^{(i+1)}\|_2^2$ is minimized over all $\alpha_0, \dots, \alpha_i$ when using the CGLS algorithm.

5. The ASTRA toolbox

There are some well-established commercial software packages to generate reconstructed images from a set of projection data (for instance Octopus Reconstruction; Vlassenbroeck et al., 2007). Their key features clearly lie in their reliability by testing over a long time period and their user-friendliness in shape of a GUI or similar. But of course there are some incisive limitations accompanied by the black-box user ansatz, for instance the small number of usable reconstruction algorithms, the fixed geometrical setup which serves to display the acquisition process of the projection data or the computational efficiency (mostly because of the usage of CPU algorithms). Similar constraints apply to current open source software tomographic reconstruction tools like PyHST2 (Mirone, Gouillart, Brun, Tafforeau, and Kieffer, 2014), TomoPy (Gürsoy, Carlo, Xiao, and Jacobsen, 2014) or TIGRE (Biguri, Dosanjh, Hancock, and Soleimani, 2016).

The All Scale Tomographic Reconstruction Antwerp (ASTRA) toolbox (van Aarle et al., 2015; van Aarle et al., 2016; Palenstijn, Batenburg, and Sijbers, 2011) has been developed since 2010 by the Vision Lab at the University of Antwerp (Belgium) to offer a new alternative software package dealing with reconstruction tasks. It aims to mix the tried and tested functionalities of commercial codes with the possibility to consult advanced developer tools and flexibilities. Therefore reasonably, the package is available as open source software for free under a GPLv3 license¹. The ASTRA toolbox routines can be called either through a MATLAB mex or a Python interface. It offers full three-dimensional flexibility to map the projection measuring process in a vectorized geometrical description. Finally, the ASTRA toolbox provides the important possibility to be run on GPUs (graphics processing units), but only NVIDIA graphic cards can be used for this purpose because the algorithms require the CUDA language. The design ideas of the ASTRA toolbox are summarized in Figure 5.1. We will go on by presenting the main components including minimal algorithmic usage examples in section 5.1. Note that all algorithmic snippets shown here refer to the MATLAB mex interface. Afterwards, we will plunge deeper into how our experimental setups from chapter 3 can be transferred to the ASTRA toolbox.

5.1. Main components of the ASTRA toolbox

The components of the ASTRA toolbox can be structured into three main concepts: depiction of the spatial geometrical setup, data handling and reconstruction algorithms. While the first two aspects further subdivide into properties of the object volume or rather of the projection composition, the algorithms of course need the full information about both the volume and the projections.

¹<https://www.astra-toolbox.com/#>

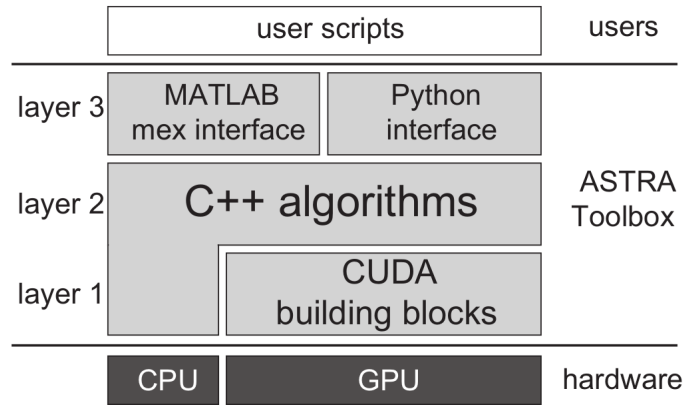


Figure 5.1.: schematic overview of the ASTRA toolbox design (taken directly from van Aarle et al., 2015, p. 36)

The input and output data have to be linked to the special measuring situation that was present when the projections have been recorded. This is why the volume and projection geometry have to be specified - they define how the volume and projection data have to be interpreted in the reconstruction algorithms later on. For the volume geometry, we have to indicate the position of the sample in the two- or three-dimensional space and even more important the pixel or voxel grid representing the object of interest.

```
% 2D volume geometry: square pixel grid centered around the origin
vol_geom = astra_create_vol_geom(num_det_cols, num_det_cols);
% 3D volume geometry: cuboid voxel grid centered around the origin
vol_geom = astra_create_vol_geom(num_det_cols, num_det_cols, num_sino);
```

Thus, each slice is a square image with $N_x \times N_x$ pixels, and in the three-dimensional case the stack is additionally N_z voxels high. The projection geometry in general comprises a keyword to identify the scanning setup (`{'parallel', 'fanflat'}` in 2D or `{'parallel3d', 'cone'}` in 3D) plus the detector pixel pitch p_x , the number of detector columns N_x and all projection angles arranged in a specific MATLAB matrix².

```
% 2D parallel beam projection geometry (p_x = 1.0)
proj_geom = astra_create_proj_geom('parallel', 1.0, num_det_cols, ...
    linspace2(0, pi, num_angles));
% 2D fan beam projection geometry (p_x = 1.0)
proj_geom = astra_create_proj_geom('fanflat', 1.0, num_det_cols, ...
    linspace2(0, 2*pi, num_angles), source_origin, origin_det);
% 3D parallel beam projection geometry (p_x, p_z = 1.0)
proj_geom = astra_create_proj_geom('parallel3d', 1.0, 1.0, num_sino, ...
    num_det_cols, linspace2(0, pi, num_angles));
% 3D cone beam projection geometry (p_x, p_z = 1.0)
proj_geom = astra_create_proj_geom('cone', 1.0, 1.0, num_sino, ...
    num_det_cols, linspace2(0, 2*pi, num_angles), source_origin, origin_det);
```

For the three-dimensional examples, the ASTRA routine also needs the detector pixel pitch p_z in e_z -direction and the number of detector rows N_z . The fan and cone beam scenarios also require storage of the exact source and detector position relative to the volume geometry. As the ASTRA toolbox understands the object sample as motionless in the origin of a fixed coordinate system, inversely the X-ray source and detector have to rotate around the origin using the sizes d_{so} and d_{od}

²Notice that the parallel beam projection geometries only need the projection angle interval $[0, \pi]$, see also the introduction of section 4.2.

and the information given in the projection angles matrix.

As a second component, data handling becomes necessary at all because ASTRA will not directly work with the standard MATLAB double precision matrices. In the case of the object volume data, we have to prepare an ASTRA data object linked to the volume geometry and initialized with zeros, which can be addressed on the user layer of the interface by a unique identifier (ID).

```
% 2D volume data
recon_id = astra_mex_data2d('create', '-vol', vol_geom, 0);
% 3D volume data
recon_id = astra_mex_data3d('create', '-vol', vol_geom, 0);
```

After we used to reconstruct image data with the algorithms included in the software package, we have to copy them back into the MATLAB workspace to possibly edit them and finally write them to image files. Again, this can be effected by making use of the unique data object ID.

```
% retrieve 2D volume data from the ASTRA data object
reconstruction = astra_mex_data2d('get', recon_id);
% retrieve 3D volume data from the ASTRA data object
reconstruction = astra_mex_data3d('get', recon_id);
```

In principle, the behaviour is the same for projection data, except for the facts that we will pass the sinogram data (which are prepared in a standard MATLAB double precision matrix) to the creational function of the ASTRA data object and no retrieve step has to be executed for this type of data, since they are only needful for the reconstruction algorithms.

```
% 2D projection data
sino_id = astra_mex_data2d('create', '-sino', proj_geom, sinogram);
% 3D projection data
sino_id = astra_mex_data3d('create', '-proj3d', proj_geom, sinogram);
```

The last main component of the ASTRA toolbox consists of the reconstruction algorithms. They subdivide into some 2D CPU algorithms (e.g. BP, FBP, SIRT, CGLS), furthermore some 2D GPU algorithms (e.g. BP_CUDA, FBP_CUDA, SIRT_CUDA, CGLS_CUDA) and lastly some 3D GPU algorithms (e.g. BP3D_CUDA, FDK_CUDA, SIRT3D_CUDA, CGLS3D_CUDA). The mathematical background of each of the specified reconstruction methods should have become clear in chapter 4. The way how to configure the algorithms is pointed out in the following code snippet.

```
% 2D FBP algorithm using the CUDA accelerated version
cfg = astra_struct('FBP_CUDA');
cfg.ReconstructionDataId = recon_id;
cfg.ProjectionDataId = sino_id;
alg_id = astra_mex_algorithm('create', cfg);
astra_mex_algorithm('run', alg_id);

% 3D SIRT algorithm including a minimum constraint condition
cfg = astra_struct('SIRT3D_CUDA');
cfg.ReconstructionDataId = recon_id;
cfg.option.MinConstraint = minconstr;
cfg.ProjectionDataId = sino_id;
alg_id = astra_mex_algorithm('create', cfg);
astra_mex_algorithm('iterate', alg_id, iters);
```

As we can see, each algorithm has to be linked to the belonging volume and projection data by their IDs. In the SIRT example, it is even possible to include the additional option of using a minimal constraint condition. If all configurations have been completed, an identifier for the algorithm object is generated. Depending on what kind of reconstruction method should be executed, the algorithm

is then ran or iterated until the resulting image data are present in the ASTRA volume data object.

5.2. Application to the experimental setups

Hypothetically, the short introduction of the last subsection should give us the opportunity to directly start a reconstruction based on a set of projection images. Nevertheless, there are some points one stumbles across by trying this. The first peculiarity occurs for 2D fan beam or 3D cone beam: When we look back to Figure 3.2 or 3.3, it is obvious that the object under investigation is enlarged onto the detector screen just like the sun casts a big shadow of a small person under certain circumstances. This effect is called geometric magnification in the field of computed tomography (cf. equation (3.1)) and quantified by the size

$$M := \frac{d_{so} + d_{od}}{d_{so}} = \frac{d_{sd}}{d_{so}}. \quad (5.1)$$

To add this factor instead of directly passing the sizes d_{so} and d_{od} to the `astra_create_proj_geom`-function, these two values must be manipulated manually to depict the situation that the detector is virtually located in the fixed origin of the coordinate system. This is expressed as follows for the two-dimensional fan beam scenario: In a first step, the detector is shifted to the origin so that

$$\tilde{p}_x := \frac{d_{so}}{d_{so} + d_{od}} p_x \stackrel{(5.1)}{=} \frac{p_x}{M} \quad \text{and afterwards} \quad \tilde{d}_{od} := 0. \quad (5.2)$$

The quantity \tilde{p}_x could also more appropriately be called the voxel size v_x (see equation (3.2)), describing the real physical dimension covered by one voxel. Secondly, all lenghts have to be rescaled to achieve $\hat{p}_x \equiv 1$, which means that the distance from the X-ray source to the iso center origin becomes

$$\hat{d}_{so} := \frac{d_{so}}{\tilde{p}_x} \stackrel{(5.2)}{=} \frac{\cancel{d_{so}}(d_{so} + d_{od})}{\cancel{d_{so}} p_x} = \frac{d_{sd}}{p_x}. \quad (5.3)$$

As the flat detector panels used for our three-dimensional cone beam scanning experiments have square-shaped detector elements with $p_x = p_z$, indeed the two manipulations from equations (5.2) and (5.3) suffice once again. What should be emphasized at this point is that from our experience, the reconstruction algorithms provided within the package were only able to give meaningful results if the projection geometry has always been created with $\hat{p}_x = \hat{p}_z \equiv 1$ after eventual lenght rescalings. This is also the reason why the code snippet in the previous section 5.1 contains these hard-coded ones in the definition of the projection geometry.

Another aspect of the ASTRA toolbox that has been withheld consciously so far is the possibility to describe the projection geometry in a vectorized way. We will point this out now for the three-dimensional cone beam scanning scenario: The projection geometry determines the location and trajectory of the X-ray source, and instead of employing a standard routine to generate the projection geometry description, we could fall back on a specific vector $\mathbf{V} \in \mathbb{R}^{N_\theta \times 12}$, where N_θ indicates the number of projection angles considered. Each row \mathbf{V}_i is assigned to one single projection angle θ_i and contains the already in chapter 3 introduced vectors $\mathbf{V}_i = (\mathbf{s}_i, \mathbf{d}_i, \mathbf{u}_i, \mathbf{v}_i)$ (used as row vectors here). To map the fact that ASTRA expects the detector to be manually placed in the iso-center of

rotation, these vectors have to be assigned to

$$\begin{aligned} \hat{\mathbf{s}}_i &= \begin{bmatrix} \sin(\theta_i) \hat{d}_{so} \\ -\cos(\theta_i) \hat{d}_{so} \\ 0 \end{bmatrix}^T, & \hat{\mathbf{d}}_i = \tilde{\mathbf{d}}_i &= \begin{bmatrix} -\sin(\theta_i) \tilde{d}_{od} \\ \cos(\theta_i) \tilde{d}_{od} \\ 0 \end{bmatrix}^T \stackrel{(5.2)}{=} \mathbf{0}^T, \\ \hat{\mathbf{u}}_i &= \begin{bmatrix} \cos(\theta_i) \hat{p}_x \\ \sin(\theta_i) \hat{p}_x \\ 0 \end{bmatrix}^T = \begin{bmatrix} \cos(\theta_i) \\ \sin(\theta_i) \\ 0 \end{bmatrix}^T, & \hat{\mathbf{v}}_i &= \begin{bmatrix} 0 \\ 0 \\ \hat{p}_z \end{bmatrix}^T = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}^T \end{aligned} \quad (5.4)$$

with the manipulated sizes from above. After such a vector \mathbf{V} has been created in the MATLAB workspace, a vectorized projection geometry can be generated like this:

```
% 3D vectorized cone beam projection geometry using predefined vector V
proj_geom = astra_create_proj_geom('cone_vec', num_sino, num_det_cols, V);
```

Figure 5.2 shows the changes of the vectors involved in the specific vector \mathbf{V} : While the starting position of the cone beam scanning scenario is indicated by \bullet_1 , another position corresponding to the projection angle θ_i (inducing the i -th row vector \mathbf{V}_i) is marked with \bullet_i . One can clearly realize from Figure 5.2 that the vectors $\{\mathbf{s}, \mathbf{d}, \mathbf{u}\}$ are varied under trigonometric functions with dependency of the specific projection angle θ_i , which is reflected in equation (5.4). In contrast, the vector \mathbf{v} pointing out of the paper plane remains invariant under increasing projection angles θ_i . Starting from such an arbitrary, rotated position \bullet_i , the detector shift to the origin \mathcal{O} like in equation (5.2) (\bullet) and the subsequent standardization from equation (5.3) ($\hat{\bullet}$) yield the input vectors $\{\hat{\mathbf{s}}_i, \hat{\mathbf{d}}_i, \hat{\mathbf{u}}_i, \hat{\mathbf{v}}_i\}$ in equation (5.4) for a vectorized ASTRA projection geometry.

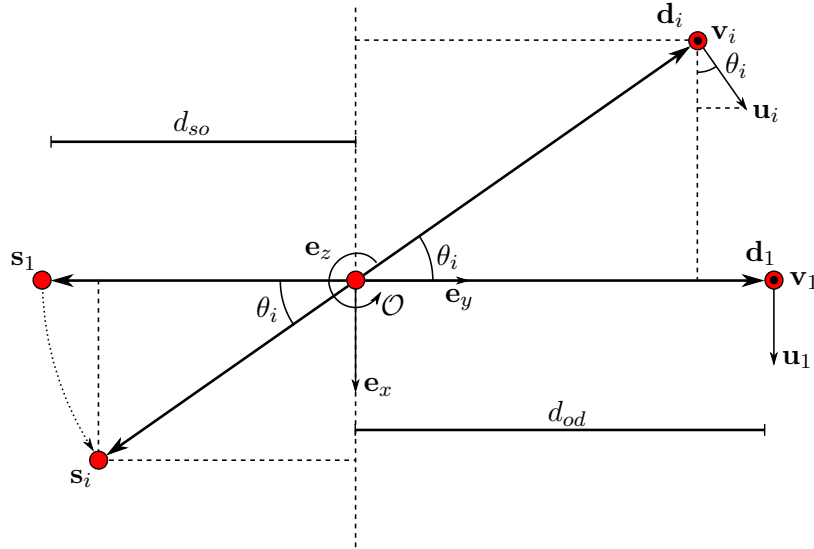


Figure 5.2.: auxiliary illustration to obtain a deeper understanding of the vectorization

To satisfy the less common reconstruction task for projections made with 3D helical cone beam scanning according to section 3.4, a few additional modifications to the vectorized cone beam projection geometry are needed. Assuming the number of entire 360° turns `num_turns` and the z -pitch per full rotation are known characteristics of the scanning process in Figure 3.4, we can calculate the z -pitch per angular change `z_trans` and the resulting total covered z -distance $z_dist = N_\theta z_trans$. The necessary projection geometry for 3D helical cone beam scanning is initialized and vectorized by means of the routines for cone beam scanning in a first step³.

```
% 3D (helical) cone beam projection geometry (p_x, p_z = 1.0)
proj_geom = astra_create_proj_geom('cone', 1.0, 1.0, num_sino_used, ...
    num_det_cols, linspace2(0, num_turns*2*pi, num_angles), ...
    source_origin, origin_det);
% generate the corresponding vectorized projection geometry
proj_geom = astra_geom_2vec(proj_geom);
```

Notice that the maximal angle argument changes by the factor `num_turns` and that the (virtual) number of detector rows has to equal the number of sinograms to be used, called `num_sino_used`. So far, the number of detector rows N_z was always identical to the total number of sinograms `num_sino` and thus the two appellations were used synonymously (compare the first code snippets in section 5.1), but this fact does not apply any more to the depiction of helical cone beam scanning. The difference will be elaborated in the context of section 7.2 later on.

The second step consists of the manipulation of the specific vector \mathbf{V} to reproduce the stepwise translation along the e_z -axis. This procedure itself is quite straightforward, but one must not forget about the standardization step to $\hat{p}_x = \hat{p}_z \equiv 1$ indicated by (\bullet) in equation (5.3). Hence, the sizes `z_trans` and `z_dist` have to be divided by the voxel size \hat{p}_z in the first instance. Afterwards, the third component of the vectors \mathbf{s}_i and \mathbf{d}_i inside the row \mathbf{V}_i is assigned to the height of the X-ray source and detector (center) along the e_z -direction for the current projection angle θ_i . To map an upward translation of the object sample, however the choice of the signs for `trans_vec` is given like in the following code snippet.

```
% adjustment of the vectorized projection geometry to 3D helical
% cone beam using the vector trans_vec
trans_vec = z_trans.*(0:(num_angles - 1))' - z_dist/2;
proj_geom.Vectors(:, 3) = proj_geom.Vectors(:, 3) + trans_vec;
proj_geom.Vectors(:, 6) = proj_geom.Vectors(:, 6) + trans_vec;
```

As the first translation of the investigated sample takes place between θ_1 and θ_2 , the z -pitch per angular change is multiplied elementwise with the vector `0:(num_angles - 1)` to create the `trans_vec`.

³The `astra_geom_2vec`-routine casts a classical into a vectorized representation of the projection geometry.

6. Image processing and filtering techniques

The ASTRA toolbox offers a wide range of functionalities concerning reconstruction techniques to the user, but it nevertheless is not a general purpose weapon. That means a user has to provide additional pre- and postprocessing steps to fulfill the expectations of the desired reconstruction images. This applies in particular to the preparational image processing from the raw projection data over normalized ones (cf. section 6.1) to the sinogram data in section 6.2 and to image filtering methods like the region of interest (ROI) mask in section 6.3. A different kind of additional functionality a user can implement is a correction of center of rotation (COR) misalignments, see section 6.4: if we take advantage of the vectorized depiction of projection geometries in the ASTRA routines, a possible experimental misalignment of the rotational axis can be dissolved. Finally, the beam hardening correction (BHC) in section 6.5 is kind of a filtering technique which remedies physical reasoned beam hardening artifacts. Similar to chapter 5, we will state code snippets where necessary.

6.1. Normalization

The normalization process of the recorded projection images is inevitable to obtain meaningful reconstruction results. Two important steps are included therein: The first processing of the projection image data was already mentioned in section 2.2 - what can be measured empirically are the X-ray intensity values, which have to be transformed nonlinearly to attenuation coefficient values by taking the negative logarithm, compare equation (2.7). Another necessary thing is to include both dark images and open beam images to adjust the projection data. Dark images (often also called dark field images) are images taken with inactive Röntgen radiation source and they indicate bad detector pixels as well as they serve as an offset correction. Contrary, open beam images (or flat field images) are recorded without the object sample when the X-ray source is turned on. These images help to rescale the projection data relative to the present surrounding medium (usually air) and to resolve defects coming from distortions in the optical path.

It is advisable to consider at least one dark and open beam image each, and if multiple ones are available, they should be averaged arithmetically before executing the subsequent code snippet¹.

```
% central element of the normalization routine
proj_norm = - log((proj - di)./(ob - di));
```

As we can see, the dark image noise `di` is subtracted from the projection data `proj` as well as from the open beam image `ob` and the result is divided by the open beam data afterwards. It turned out that another scaling of the maintained attenuation coefficients is also significant: The normalized data should be rescaled under usage of their global minimal and maximal values directly after the normalization itself. Thus, we need to iterate over all normalized projection images in a first loop of a possible implementation to ascertain and save the minimum and maximum. In a second iteration, each pixel value of the normalized projection images is rescaled to the interval $[0, 1]$.

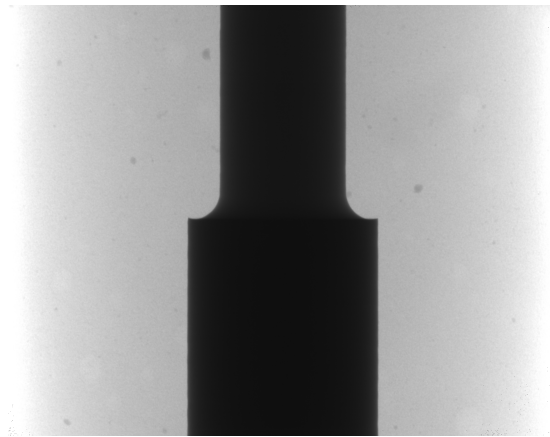
¹The MATLAB `log`-operation does not take the decadic logarithm, as one could suspect, but the requisite natural logarithm $\ln(\bullet)$. The `proj`-values match the intensity $I(L)$ and the `ob`-values the intensity I_0 from equation (2.7).

```
% rescaling with known global minimal and maximal pixel value
proj_norm_resc = (proj_norm - min)./(max - min);
```

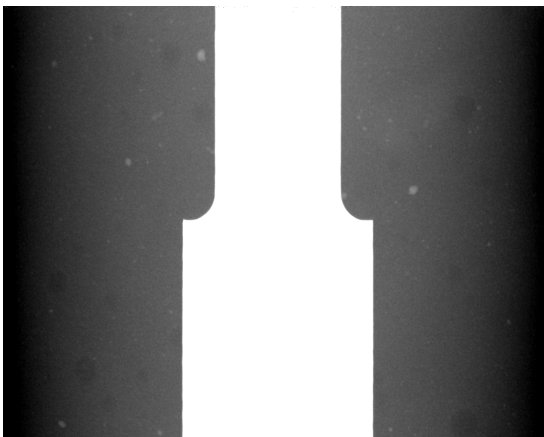
The resulting image data contained in the double precision matrix `proj_norm_resc` can then be written to image files of desired output format (e.g. 16-bit TIFF) using MATLAB standard routines. Figure 6.1 primes the normalization routine with an application example: The raw, visually unedited projection image of a stepped brass cylinder phantom in Figure 6.1(a) (which has been rescaled for better visibility in Figure 6.1(b)) is normalized first, which would yield the saved output image in Figure 6.1(c). As clearly visible, an identical gray value is assigned to the entire cylinder, which is an undesirable side effect of image data type conversions in the MATLAB reading and writing routines (all double values of the normalized image larger than 1 are cut off to the maximal 16-bit unsigned integer value of 65535). It is therefore compulsory to rescale the normalized image as previously described, resulting in an output like in Figure 6.1(d).



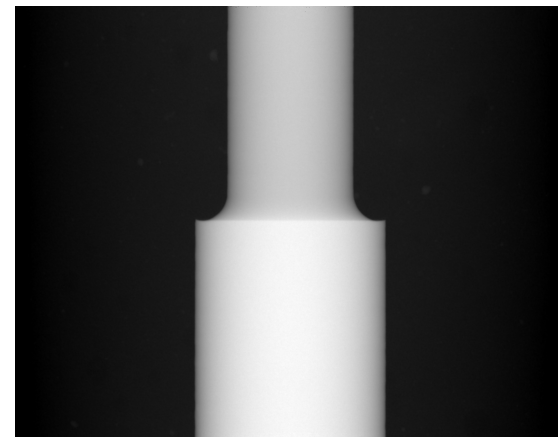
(a) raw input image `proj`



(b) rescaled input image (just for illustration)



(c) normalized image `proj_norm`



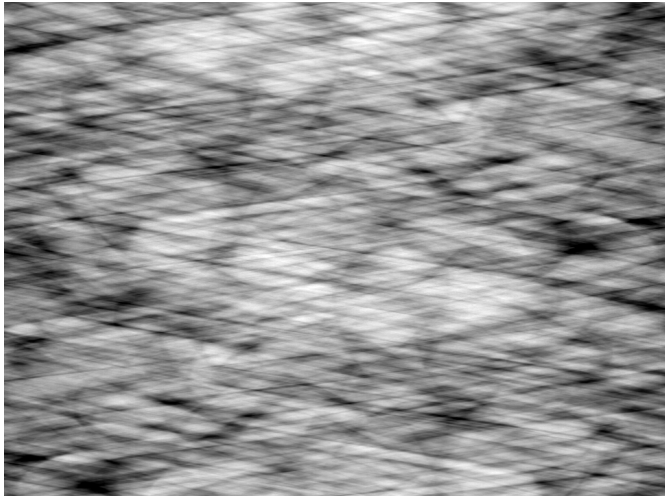
(d) normalized and rescaled image
`proj_norm_resc`

Figure 6.1.: normalization steps for one projection image of a stepped brass cylinder phantom with

$$M^{(5.1)} = 16.50$$

6.2. Sinograms

The creation of sinogram images does not comprise any kind of pixel value editing, but is rather a rearrangement procedure of the projection data (which can, but does not need to be preprocessed by normalization according to section 6.1) in a specific, problem adapted order. So far, if we assume that a set of normalized and rescaled projection images is given, each projection image has the size $N_z \times N_x$, equal to the resolution of the flat detector panel for a parallel or cone beam scanning scenario. The entirety of projection images counts N_θ many individual images, where N_θ again stands for the number of projection angles of the measuring process. The transformation to sinogram images runs as follows: We fix one single detector row, which consists of N_x pixel values, and track the gray value trend of this particular row over all available projection angles N_θ . Thus, each single sinogram includes $N_\theta \times N_x$ pixel values, and altogether the set of sinograms is obviously N_z large, where N_z is the number of detector rows. To give an impression of real-world sinograms, Figure 6.2 shows two sinograms for two different object samples recorded with various scanning setups. Figure 6.2(a) contains an asphalt sinogram measured with cone beam scanning for $N_\theta = 1440$ projection angles. Figure 6.2(b) however depicts a sinogram for the stepped brass cylinder phantom already used in subsection 6.1 (for another geometric magnification), this time measured using helical cone beam scanning with $N_\theta = 2880$ projection angles. The distinct difference between the two scanning scenarios is that the object itself moves vertically through the detector rows for helical cone beam scanning, producing a sinogram like the one in Figure 6.2(b).



(a) cone beam scan of an asphalt sample with $M^{(5,1)} \approx 2.99$
($N_\theta = 1440 \times N_x = 1944$ pixels)



(b) helical cone beam scan of a stepped brass cylinder phantom with $M^{(5,1)} \approx 5.98$
($N_\theta = 2880 \times N_x = 830$ pixels)

Figure 6.2.: exemplary sinograms generated from corresponding normalized projection images

One could ask the question what the advantage of the whole rearrangement should be. The reason for this lies in an implementation aspect: For one slice to be reconstructed with algorithms from the ASTRA toolbox (exemplarily), which has the dimensions $N_x \times N_x$, the sinogram corresponding to this detector row (belonging to the resulting slice) contains all projection data required for a reconstruction algorithm. To tell the whole truth, this substantiation is only valid for parallel beam projection geometries in 2D or 3D and the fan beam scanning scenario in 2D, as in these cases one X-ray beam passes straightforward through the object in a slice-wise manner before impinging orthogonally on the detector. Therefore, one sole sinogram includes all necessary information to reconstruct the penetrated sample slice - the reason why the transfer from projection to sinogram data is widespread in computed tomography. By the way, the naming "sinogram" of this depiction method refers to measurements of a point-like object which is positioned eccentric from the axis of rotation: if the projections of this sample are rearranged to a sinogram, the viewer will observe a sinusoidal graph along the image rows serving as an ordinate. This can for instance be observed well in Figure 4.5(b).

6.3. Cropping and region of interest (ROI) mask

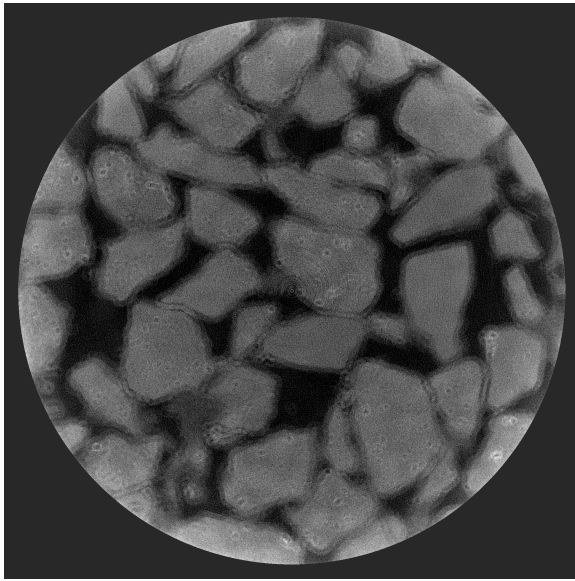
In this subsection, two minor image processing methods are merged because they both restrict the image data in a certain manner. The first one will be named "cropping" here and is used in the following situation: Normally, the input projections are measured with a geometric magnification sufficient to record the full object sample including some blank space (usually covered by the surrounding air) to guarantee that the object is always located inside the field of view (FOV). Then cropping denotes the editing of projection images by cutting off some pixels which are not of interest for the investigator, for example the blank space pixels. One has to take care when performing the cropping that the region the user wants to remove is irrelevant for the whole set of projections, which is possibly not the case for a misaligned center of rotation (compare section 6.4) or an asymmetrical object of interest. We applied the cropping routine after the normalization step in section 6.1, but there should be no difference in which order this is done as long as the cropping is finished before starting the reconstruction algorithm. The clear advantage of cropping is the (noticeable) reduction of memory space needed for the projection images and each image subsequently saved as part of the whole reconstruction procedure.

The region of interest (ROI) mask however is a postprocessing method which is applied to the reconstructed images. After utilizing one of the ASTRA reconstruction algorithms, each resulting slice has the dimensions $N_x \times N_x$, which was already mentioned in section 5.1. Thereof, only the circle around the central pixel $(N_x/2, N_x/2)^T$ with radius $N_x/2$ contains the relevant object data. All pixels outside this circle show quite arbitrary values, which are artifacts of the reconstruction algorithm used, do not have any physical meaning and especially disturb the contrast ratio of the output pictures. This problem is also independent of the reconstruction algorithm employed and must therefore be solved with the additionally implemented ROI mask function: Firstly, the user shall define a percentage ratio of the radius to enclose the circle of valuable reconstructed values. Per default, this value is set to 100 %, but often a lower ratio (for example 95 %) is advisable. The reason for this is that with increasing radial distance from the central pixel there are fewer projection information available per pixel, leading to worse reconstruction data if the number of projection angles stays unchanged. If the ratio has been set, all pixel values outside the just mentioned circle are replaced by any permissible gray value. The following code snippet demonstrates how the key functionality of the ROI mask is implemented in our case.

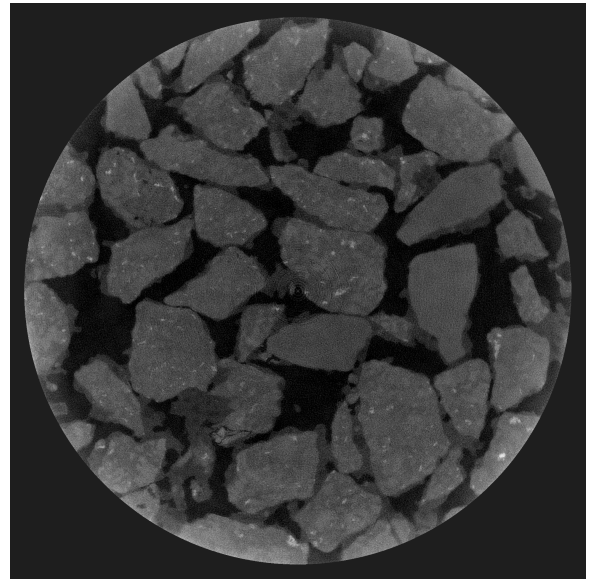
```
% ROI mask for 3D reconstruction data handed in rec_matrix
for i = 1:size(rec_matrix, 3)
    rec_matrix(:, :, i) = rec_matrix(:, :, i).*circ_matrix + ...
        (rec_mean/2).*(circ_matrix == 0);
end
```

The variable `circ_matrix` is a $N_x \times N_x$ large logical matrix in which each pixel position inside the defined circle is associated with a `true`-value. As you can see, all reconstructed image pixels inside the circle remain unchanged, whereas all pixels outside are replaced with half of the global mean gray value of all reconstruction data `rec_matrix` (this choice was made for optical purposes only).

6.4. Correction of center of rotation (COR) misalignments



(a) horizontal COR misalignment by 10 pixels



(b) optimally chosen COR parameters

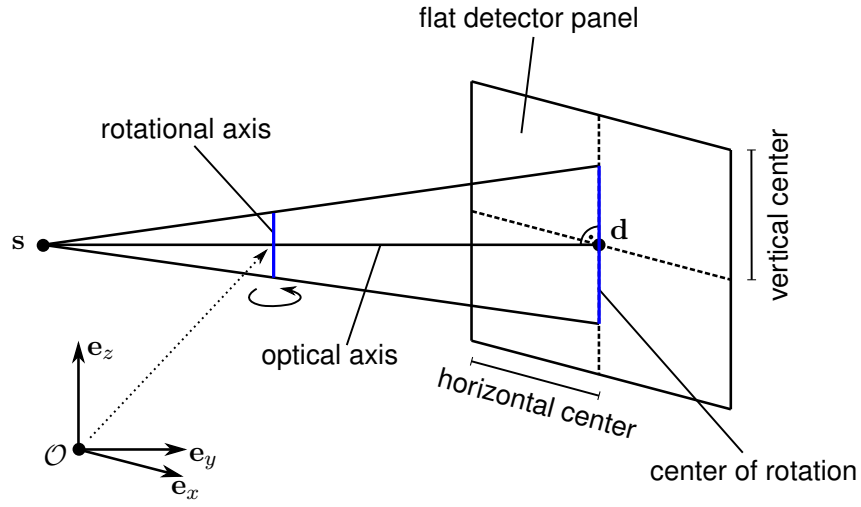
Figure 6.3.: reconstructions of an asphalt sample with $M^{(5.1)} = 2.99$ for different COR parameters

A very important feature which is not yet mapped as a standard ASTRA toolbox routine is the possibility to compensate some center of rotation (COR) misalignments mainly induced by suboptimal experimental projection recording. For non-vectorized projection geometry configurations of the ASTRA toolbox, such a correction is impossible without touching the lowest implementation level of the involved reconstruction algorithms, and also for vectorized projection geometries the user has to concern himself with a suitable implementation. Misaligned CORs lead to undesirable artifacts like double edges and therewith to a significant deterioration in the image quality of the reconstruction (see for example Louk and Suparta, 2015). Figure 6.3(a) gives an example for a reconstruction (starting from exactly the sinogram in Figure 6.2(a) for the asphalt sample) that was carried out with an uncorrected COR and underlines the need of such an algorithmic option - otherwise meaningful results like the one in Figure 6.3(b) remain unachievable. Such a malposition arises if the rotational axis of the experimental setup is not perfectly projected onto the middle of the flat detector panel (for further details, look at Ferrucci, Leach, Giusca, Carmignato, and Dewulf, 2015). In other words, an ideal alignment would be given if the projection values of the vertical center line through the sinogram equal the attenuation values of the rotational axis. As even a difference in subpixel precision can lead to COR misalignment artifacts, this interference can not be eliminated by the experimenter

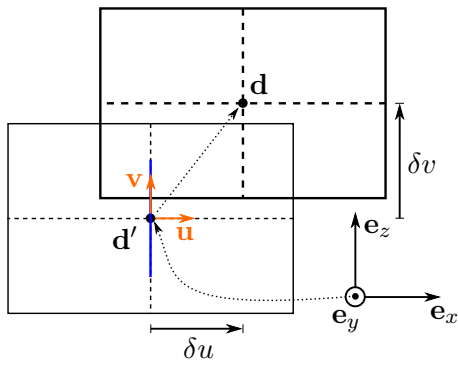
through means of the used hardware, but has to be removed inside the computation of the reconstruction. In the following subsection 6.4.1, we will focus on a more detailed explanation and depiction of possible error sources. After that, subsections 6.4.2 and 6.4.3 will show how a COR shift respectively tilt can be handled both mathematically and in the implementation.

6.4.1. Characterization of possible center of rotation (COR) misalignments

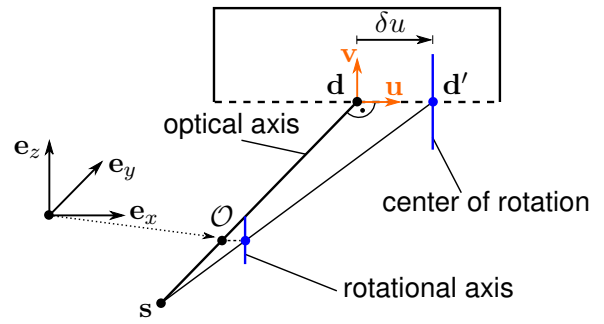
Figure 6.4(a) clarifies the situation of a perfectly aligned experimental setup and incidentally introduces the identifiers for the current problem (see also “Octopus Reconstruction User Manual”, 2018). The fixed Cartesian coordinate system is located at the intersection point of the optical axis with the rotational axis of the ideal measuring situation (and is also kept like this for the misaligned constructions in Figures 6.4(b) - 6.4(e)). The optical axis is characterized through the fact that the corresponding X-ray beam hits the flat detector panel surface orthogonally. With ideal orientation, this specific intersection point (yielding the horizontal and vertical center) coincides with the middle of the detector, where the vector \mathbf{d} also points to. Furthermore, the center of rotation (COR) line is defined by the projection of the rotational axis onto the detector panel. If the COR only covers the single detector column lying at the horizontal center, again the measurement scenario was constructed optimally. The first deviations from a perfect lining up are given in Figure 6.4(b) and Figure 6.4(c) and are called COR shift: In the first case depicted in Figure 6.4(b), the detector panel's center \mathbf{d} was displaced both horizontally by δu and vertically by δv , while the position \mathbf{d}' where the optical axis impinges is kept unmodified. The same situation can also be obtained by a displacement of the X-ray source \mathbf{s} and optical as well as rotational axis in the opposite direction with fixed detector position (and coincides anyway in the mathematical description, except for signs). The second and more common case in Figure 6.4(c) develops if the rotational axis is displaced horizontally by the amount δu from the optical axis going through the origin \mathcal{O} of the Cartesian coordinate system, while the X-ray source \mathbf{s} and detector \mathbf{d} stay unaffected. One can clearly see that the center of rotation does not pass through the horizontal center of the detector panel any more. Nevertheless, both COR shifts will be treated in the same manner in subsection 6.4.2 because changes in the angle of impact for the affiliated X-ray beams regarding the positions \mathbf{d} and \mathbf{d}' are neglected. The situation portrayed in Figure 6.4(d) comes from a detector respectively rotational axis distortion and will be called a COR tilt misalignment from now on. To be exact, Figure 6.4(d) implies a tilt of the rotational axis by the angle φ in mathematical positive direction with respect to the \mathbf{e}_y -axis of the global coordinate system. Besides that, the key characteristic of a COR tilt is that the COR does not fall onto one single detector column, but onto several ones. Finally, Figure 6.4(e) portrays a COR skew generated by the rotation of the detector by the skew angle χ with respect to the \mathbf{e}_x -axis of the fixed Cartesian coordinate system. As we will not deal with this kind of misalignment in the following because of its very little influence on the image quality of the reconstruction, this has only been specified for the purpose of completeness. Nevertheless, the correction of a COR skew can easily be derived with the knowledge from subsection 6.4.3 concerning a COR tilt.



(a) undisturbed experimental (cone beam) scanning setup including labels



(b) COR shift caused by displacement of the detector's center



(c) COR shift caused by horizontal displacement of the rotational axis from the optical axis

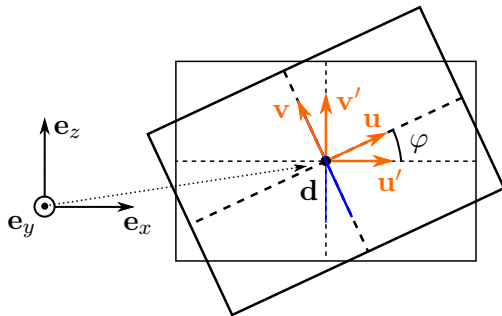
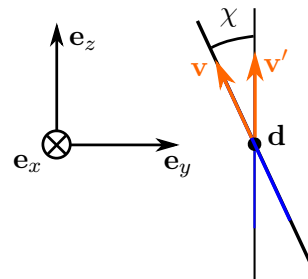
(d) COR tilt by angle φ (e) COR skew by angle χ

Figure 6.4.: overview of possible center of rotation (COR) misalignments

6.4.2. Correction of misalignments through center of rotation (COR) shift

The most powerful tool of COR misalignment correction is undoubtedly dissolving misalignments through a COR shift. The impacts of a vertical or horizontal displacement have already been pointed out in the introductory part and also in Figure 6.4(b) or 6.4(c), so we will directly pass over to the description of the correction in mathematical terms. The vectors $\{\mathbf{s}, \mathbf{u}, \mathbf{v}\}$ remain unmodified, while the COR shift correction is carried out via

$$\mathbf{d}' = \mathbf{d} - \delta u \mathbf{u} - \delta v \mathbf{v}. \quad (6.1)$$

For given displacements $\delta u, \delta v$ the ASTRA projection geometry can be corrected as specified in the following code snippet.

```
% correction of COR shift for 3D vectorized cone beam projection geometry
proj_geom.Vectors(:, 4:6) = proj_geom.Vectors(:, 4:6) - ...
    delta_u.*proj_geom.Vectors(:, 7:9) - ...
    delta_v.*proj_geom.Vectors(:, 10:12);
```

Remember how the vector \mathbf{V} has been assembled in section 5.2 when the possibility to use a vectorized projection specification was introduced to understand that equation (6.1) can immediately be applied in an implementation. What should also be emphasized here is that **any** kind of COR misalignment can only be corrected in the ASTRA toolbox if the vectorized form of the projection geometries is used. One big negative side effect thereof is that this fact reduces the number of usable reconstruction algorithms to the ones which can be called with vectorized projection geometries (eliminating for instance the prominent analytical reconstruction algorithms FBP in 2D and FDK in 3D), since the COR correction explicitly needs this requirement. For example, the reconstruction of the asphalt sample with optimal horizontal COR parameter δu in Figure 6.3(b) was generated using the CGLS_CUDA algorithm with 50 iteration steps.

6.4.3. Correction of misalignments through center of rotation (COR) tilt

The second correction we will treat here is the one of a tilted COR in the form portrayed in Figure 6.4(d). If for instance the rotational axis has a tilt of tilt angle φ , this does not really affect the slices in the middle of the stack of reconstructed images corresponding to the vertical center (since usually $\varphi < 1^\circ$, and definitely not as large as in the illustration). But indeed a COR tilt correction can help to improve the quality of the reconstruction results in the outer regions along the \mathbf{e}_z -direction because the COR line will fall onto some non-central detector columns there. If we define the rotational matrix \mathbf{R}_φ as

$$\mathbf{R}_\varphi := \begin{bmatrix} \cos(\varphi) & 0 & -\sin(\varphi) \\ 0 & 1 & 0 \\ \sin(\varphi) & 0 & \cos(\varphi) \end{bmatrix}, \quad (6.2)$$

which gives the “forward” mapping $\{\mathbf{u}' \xrightarrow{\mathbf{R}_\varphi} \mathbf{u}, \mathbf{v}' \xrightarrow{\mathbf{R}_\varphi} \mathbf{v}\}$, the COR tilt correction simply is

$$\mathbf{u}' = \mathbf{R}_\varphi^T \mathbf{u} \quad \text{and} \quad \mathbf{v}' = \mathbf{R}_\varphi^T \mathbf{v} \quad (6.3)$$

with unchanged source vector \mathbf{s} and detector vector \mathbf{d} . Of course we used the property of a rotational matrix that its inversion can be expressed through its transposing operation, leading to the form of equation (6.3). As already mentioned, a rotation matrix for the skew angle χ similar to the one presented in equation (6.2) solves the challenge of a COR skew (see Figure 6.4(e)) misalignment. If the tilt angle φ and the corresponding rotational matrix \mathbf{R}_φ have been predefined, the ASTRA projection geometry is again manipulated exactly as described in equation (6.3).


```
% correction of COR tilt for 3D vectorized cone beam projection geometry
for i = 1:size(proj_geom.Vectors, 1)
    proj_geom.Vectors(i, 7:9) = proj_geom.Vectors(i, 7:9)*rot_matrix;
    proj_geom.Vectors(i, 10:12) = proj_geom.Vectors(i, 10:12)*rot_matrix;
end
```

A little tricky herein is the multiplication with the (untransposed) rotational matrix \mathbf{R}_φ from the right: As the vectors \mathbf{u} and \mathbf{v} are saved as row vectors inside the global vector \mathbf{V} (compare section 5.2), performing the operation $\mathbf{R}_\varphi^T \mathbf{u}$ on a column vector \mathbf{u} (which was assumed in the mathematical formulation above) can identically be done with $\mathbf{u} \mathbf{R}_\varphi$ if only a row vector \mathbf{u} is available. The same is obviously valid for the row vector \mathbf{v} , and this difficulty is also the reason for the loop presented in the code snippet.²

6.5. Beam hardening correction (BHC)

The last image filtering technique presented in this section is the beam hardening correction (BHC; Carmignato et al., 2018, p. 165 ff.). The physical background is given by the law of Beer-Lambert, as it already has been presented in section 2.2. If we recap equation (2.7),

$$I(L) = I_0 e^{-\int_0^L \mu(x) dx} \Leftrightarrow p = \int_0^L \mu(x) dx = -\ln(I(L)/I_0),$$

we can see that one specific projection value p is found by integration of the attenuation coefficient $\mu(x)$ along the corresponding X-ray beam path only using its spatial dependency. This is indeed just a simplified version and does not take into account that the attenuation coefficient is also energy-dependent, meaning $\mu = \mu(x, E)$. As we will see, the form of the law of Beer-Lambert in equation (2.7) is not sufficient any more to illuminate beam hardening artifacts coming from the polychromatic nature of Röntgen radiation (recap the X-ray spectra in Figure 2.2 for an illustration of the polychromatic properties). If we additionally include that the X-ray source emits an initial spectrum $I_0(E)$, Beer-Lambert's law in equation (2.7) can be extended to a non-linear version

$$I(L) = \int_0^{E_{max}} I_0(E) e^{-\int_0^L \mu(x,E) dx} dE. \quad (6.4)$$

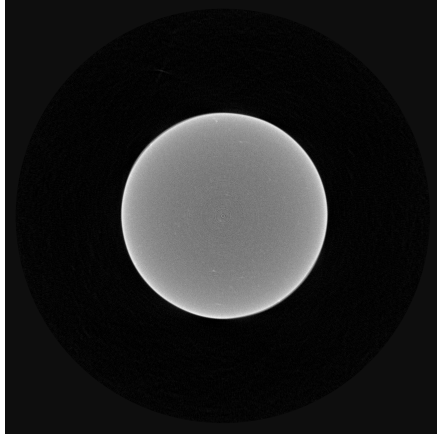
Expanding the definition of the incident intensity value to $\tilde{I}_0 = \int_0^{E_{max}} I_0(E) dE$, the polychromatic, non-linear reshaping of equation (6.4) reads

$$p = -\ln \left[\underbrace{\frac{1}{\tilde{I}_0} \int_0^{E_{max}} I_0(E) e^{-\int_0^L \mu(x,E) dx} dE}_{=I(L)} \right]. \quad (6.5)$$

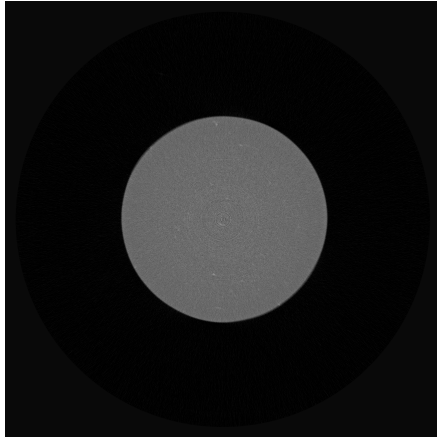
This nonlinearity is usually not considered in the reconstruction process as the intensities can not be differentiated in distinct energies, and this in turn leads to beam hardening effects in reconstructions based on equation (2.7): If a broad-band energy spectrum $I_0(E)$ passes through an object, its spectrum changes along the path since varying frequency bands of the spectrum are attenuated differently, depending on the attenuation coefficient $\mu(x, E)$ of the specific sample material, compare equation (6.5). In general, the low-energy (soft) X-ray beams are more strongly attenuated

²Unfortunately, the presented COR tilt correction could not be validated yet in practice for lack of time.

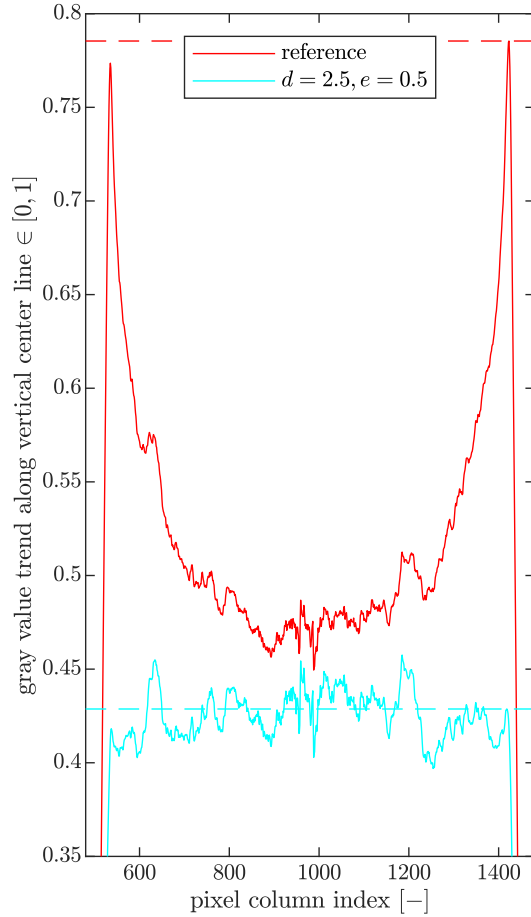
than the high-energy (hard) ones. This fact generates the so-called beam hardening artifacts in the reconstruction images, which are of course an undesirable side effect of the assumption that the X-ray source produces monochromatic radiation.



(a) reference reconstruction without BHC



(b) reconstruction with "optimal" BHC fitting parameters



(c) numerical study of the gray value trend (designated coefficients cf. equation (6.6))

Figure 6.5.: comparison of reconstructions for the small diameter region of a stepped brass cylinder phantom ($M^{(5.1)} = 33.02$) with regard to the application of BHC

The result of beam hardening can be identified as following (compare Stock, 2008): The accessible projection values p underestimate the attenuation coefficients, leading to a lowering opacity along increasingly long X-ray paths which suffer the most from this underestimation. Consequently, the material absorption decreases with increasing depth of the penetrated material - yielding smaller intensities in the center of the reconstructed images than their true attenuation value would be. This beam hardening artifact is possibly better known under the name "cupping effect", whose naming becomes understandable with a look at Figure 6.5: The two images on the left both show reconstructions of the stepped brass cylinder phantom which has already been mentioned several times (the portrayed slice comes from the region with smaller diameter, cf. Figure 6.1). While Figure 6.5(a) embodies the reference reconstruction with clear cupping effect visible, the result of a reconstruction using the BHC method explained in the following with an "optimal" choice of fitting parameters can be found in Figure 6.5(b). Additionally, Figure 6.5(c) plots the actual gray data values along the cutting line at half image height, restricted to pixel column indices near the location of the cylinder

for a better representation. The dashed lines each symbolize the maximum of the cylinder border values in the reconstructed images. An optimal reconstruction taking the beam hardening effect into account would be given if the gray values were constant along the path length through the object - and the parameters in the legend of Figure 6.5(c) can approach this behaviour well since they oscillate around the dashed cyan line.

To remove this kind of artifacts, a BHC could be done both utilizing additional hardware in the measurement process and exploiting a software solution with prior knowledge of the sample's material properties. Of course we only take a closer look at the computational correction here. One ansatz consists of a linearization technique in which polynomial curves for the projection gray values are parametrized to reduce the cupping effects in the final reconstruction (the representation below is taken from Carmignato et al., 2018, p. 173 f.). The very simple correction is directly implemented in the reconstruction software in the shape

$$Y = a (b + cX + dX^2 + eX^3 + fX^4 + gX^5) . \quad (6.6)$$

Herein, X is the initial gray value of a single pixel in the projection images (after normalization, see section 6.1), Y on the other hand the BHC-adjusted, linearized gray value and $\{a, b, c, d, e, f, g\}$ the set of fitting parameters of the polynomial curve given by equation (6.6). The simplicity of the BHC is at the expense of the manual user-driven selection of the parameters in a trial-and-error way. Nevertheless, it is an adequate method to cause a noticeable amelioration both for materials with low absorption characteristic as well as for higher absorbing materials (as given in Figure 6.5) or even multi-component objects of interest. The choice of the coefficients in equation (6.6) is crucial to achieve this goal. There are also some iterative methods to correct for beam hardening artifacts, but these are significantly larger than the scope of this work.

To investigate the influence of the BHC method proposed in equation (6.6) systematically, a numerical study for the already presented reconstructed slice has been carried out, whose outcome is pictured in Figure 6.6. Although equation (6.6) also permits the usage of the coefficients $\{a, b, c\}$, this has not been examined because a change up to the first monom has not been considered useful. This reflection was approved by a look into the user manual ("Octopus Reconstruction User Manual", 2018) of the Octopus Reconstruction software³, in which a simple polynomial BHC method is contained, too. That means we kept the lower coefficients at $\{a = 1.0, b = 0.0, c = 1.0\}$ and raised the magnitude of each higher fitting parameter $\{d, e, f, g\}$ independently of each other. Generally speaking, each parameter can contribute to an improvement of the resulting reconstruction image. The increment in Figure 6.6(a) for the d -coefficient has a higher magnitude because the influence of this coefficient on the gray values is not that high, compared to the higher-order coefficients in Figures 6.6(b) - 6.6(d). Taking each diagram for itself, an individual (manual) choice of the belonging fitting parameter can be identified by analyzing the gray value trend with respect to the distance to its dashed border value line. For the stepped brass cylinder phantom example, Figure 6.6 would suggest the individual choices $\{d = 3.0, e = 1.0, f = 1.0, g = 1.5\}$. But of course the superposition of all these fitting parameters clearly overestimates the desired gray value trend, so a well-suited BHC can be found with a moderate choice like $\{d = 2.5, e = 0.5\}$, as it was made for the "optimal" reconstruction in Figure 6.5. Another cognition especially of Figures 6.6(c) and 6.6(d) is that the coefficients corresponding to higher polynomial orders amplify the gray values of the central pixel column indices unreasonably much.

³<https://octopusimaging.eu/>

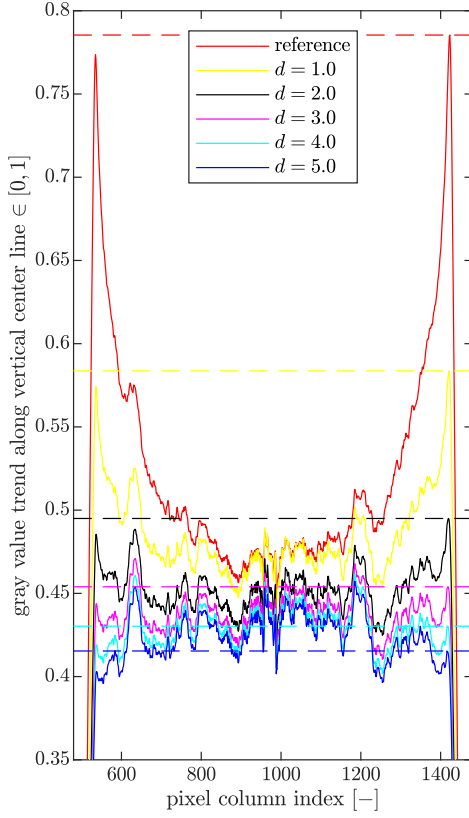
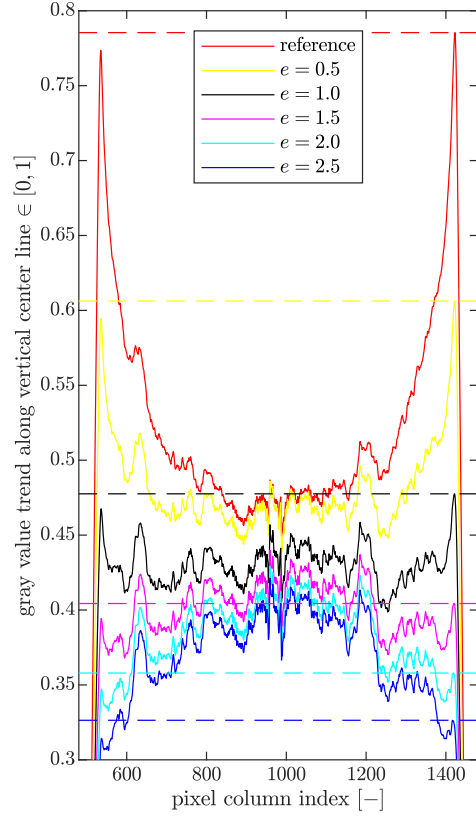
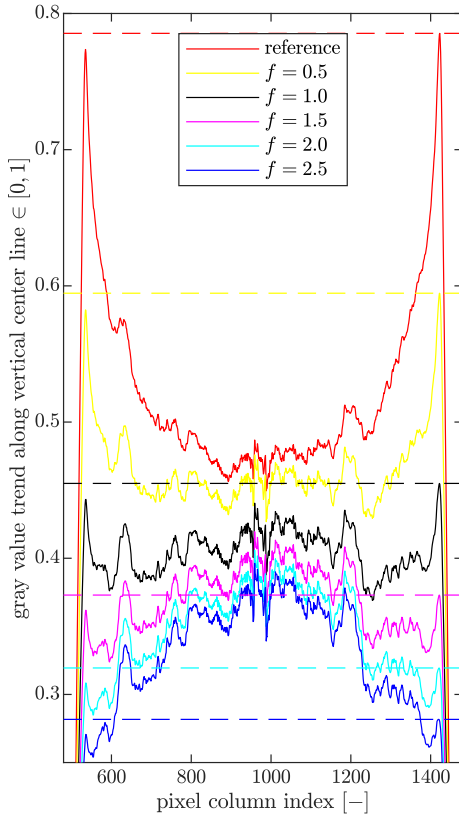
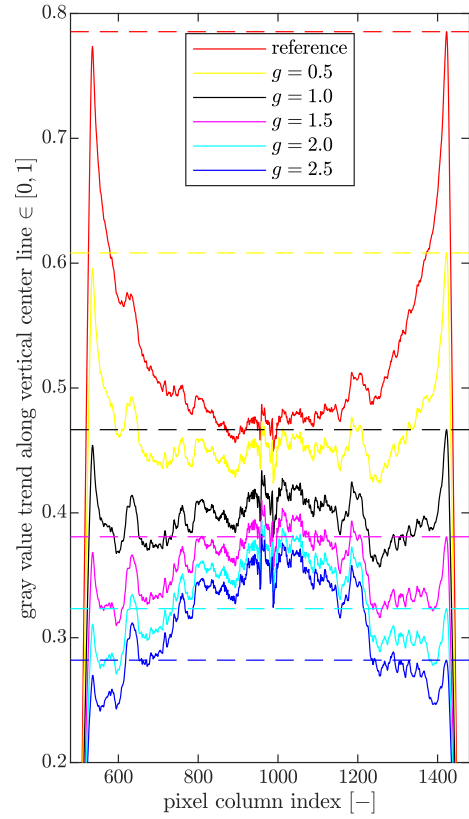
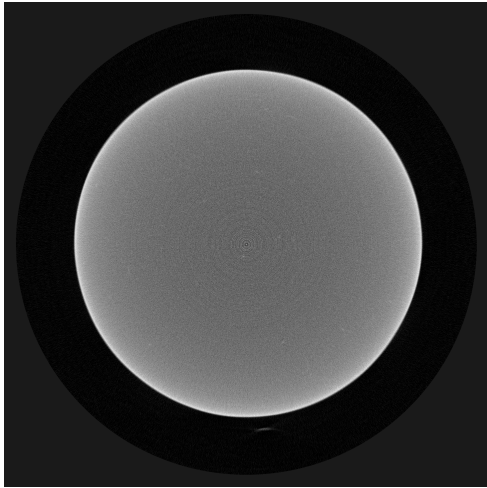
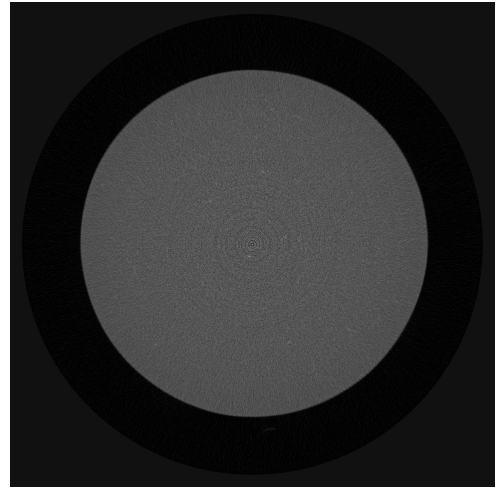
(a) coefficient d with 1.0-increment(b) coefficient e with 0.5-increment(c) coefficient f with 0.5-increment(d) coefficient g with 0.5-increment

Figure 6.6.: numerical studies of the gray value trend for each coefficient in $\{d, e, f, g\}$ independently, compared to the reference reconstruction in Figure 6.5(a) with unchanged overall setting



(a) reference reconstruction without BHC



(b) reconstruction with "optimal" BHC fitting parameters

Figure 6.7.: comparison as in Figure 6.5 for the large diameter region

To finish the investigation, the detected (good) choice of the fitting parameters $\{d = 2.5, e = 0.5\}$ was transferred to another region of the stepped brass cylinder phantom with larger diameter (for constant geometric magnification) in Figure 6.7. The proposed simple polynomial BHC method produces a good result for the reconstructed slice related to the larger diameter region (cf. Figure 6.7(b)), just as required.

7. Results, comparison and discussion

The issue of this penultimate chapter will be if all the preparations made in chapters 4 through 6 suffice to conduct good reconstructions for practice-oriented cone or helical cone beam scanning data sets. We can anticipate that this is not the case, and so section 7.1 explicates with which additional methods it was possible for us to accomplish good reconstruction results for an irregular real-world data set measured by cone beam scanning. This section is rounded off with the actual fulfillment of the thesis' objective to be able to reconstruct full three-dimensional μ XRCT cone beam scanning data sets. In contrast, section 7.2 discusses to what extent we were able to treat helical cone beam scanning data and which extra considerations play a major role on the way to a reconstruction for this scan setup.

7.1. Processing of a full cone beam scanning data set

We want to approach the goal of this thesis slowly, which is to offer the opportunity to reconstruct a full experimental (cone beam scanning) data set detected in the open, modular and flexible in-house XRCT system of the Institute of Applied Mechanics (CE) at the University of Stuttgart. In theory, we should be able to achieve this offhand using the whole preparations from chapters 5 and 6. However, there was still a big obstacle in the way of the practical enforcement: the amount of data. To run the GPU-accelerated ASTRA reconstruction algorithms, the toolbox transmits the necessary sinogram data into the allocatable GPU RAM where it is held available for the algorithmic operations. This is uncomplicated for one single sinogram (for instance in 2D reconstruction) with a typical size of 1440×1944 pixels¹, each encoding a 16 bit gray value information which is saved internal as an 8 byte double precision value in MATLAB², consuming approximately 22.39 MB in total. But as a full 3D cone beam data set (exemplary) embraces 1536 sinograms, the required GPU RAM amounts to about 34.4 GB data volume, not yet including any further free memory for workspace variables of the computation. These hardware requirements will commonly not be available, as it was the case with us, and so the necessity of partitioning pops up.

This directly raises the question of how a sensible partitioning should look like. One has to consider that one single X-ray beam passes multiple voxel layers (the number depends on the location inside the entire cone), corresponding to several sinograms. Viewed inversely, we have to estimate how many sinograms have to be present in the GPU RAM simultaneously for the reconstruction of one slice. In the worst case for the outermost X-ray of the cone, the necessary slice count can be identified as

$$\text{nec_slice_count} = \left\lceil \frac{\Delta h_{ray}}{\tilde{p}_z} \right\rceil = \left\lceil \frac{\tan(\omega) N_x \tilde{p}_x}{\tilde{p}_z} \right\rceil$$

by some easy geometrical deliberations according to Figure 7.1. This expression can be simplified further by replacing $\tan(\omega) = N_z p_z / 2d_{sd}$ (regard the big triangle in Figure 7.1 for this) and convert-

¹This is the resulting standard size for many scans executed with the Dexela 1512NDT X-ray detector of the in-house XRCT lab, refer to section 2.3 for more details over it.

²The choice of the data format is linked to the allowed formats for the initialization routine of the ASTRA sinogram data objects, cf. section 5.1. The `astra_mex_data_2d/3d` function expects a MATLAB matrix of class double, single or logical.

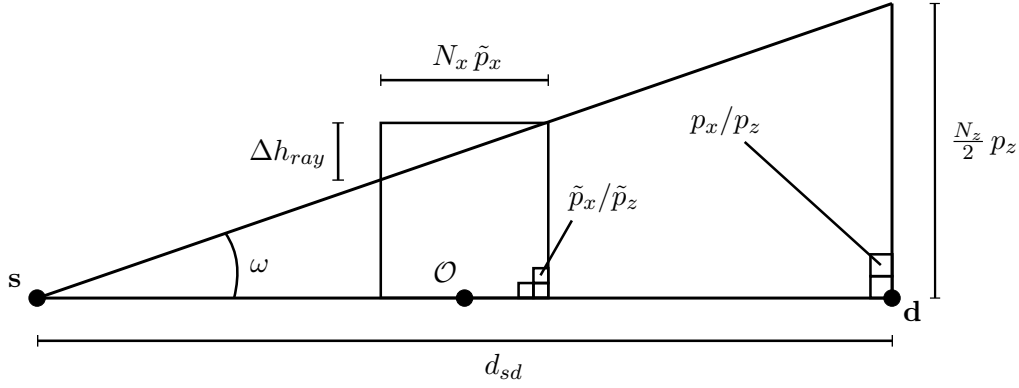


Figure 7.1.: auxiliary sketch of the geometrical situation for the definition of `nec_slice_count`

ing $\tilde{p}_x = p_x/M$ and $\tilde{p}_z = p_z/M$ each with a look back to equation (5.2). Hence, the final correlation for the necessary slice count is given by

$$\text{nec_slice_count} = \left\lceil \frac{N_x N_z p_x}{2d_{sd}} \right\rceil. \quad (7.1)$$

In our case, this size can be seen as a constant because the choice of a specific detector dictates the numerator sizes and the distance from the X-ray source to the detector d_{sd} is invariable in the XRCT lab setup. As even the calculated number $\text{nec_slice_count} = 113$ from equation (7.1) did not fit into the GPU RAM, we were finally satisfied with $\lceil \text{nec_slice_count}/2 \rceil = 57$.

7.1.1. Introduction and utilizability of the overlap

Unfortunately, a subordinate problem came along with the partitioning: the marginalized slices of such an individual partition exhibit poor quality of the 3D reconstruction (which is a frequent phenomenon because of the shrinking number of projection values disposable in the border areas). This means a full reconstructed data set reveals periodical zones of poor image quality, exactly at the positions where the changes between two partitions take place. To cope with this additional challenge, we inserted an overlap region in between the transition of two adjacent partitions, called “the overlap” from now on. The exact procedure is presented in the following paragraph.

The first two steps actually belong to the partitioning procedure itself. As a start, we have to calculate the theoretically requisite number of partitions via

$$\text{partitions} = \left\lceil \frac{N_z}{\text{nec_slice_count}} \right\rceil^3. \quad (7.2)$$

This yields $\text{partitions} = 26$ for the above-mentioned example with the reduced number of necessary slices of 57. Secondly,

$$\text{num_slices_per_part} = \left\lceil \frac{N_z}{\text{partitions}} \right\rceil \quad (7.3)$$

gives the actual number of slices per partition that will be reconstructed all at once. The rounding operations in equations (7.2) and (7.3) ensure that the global algorithm will reconstruct the full data

³If the hardware constraints are very strict, the usage of the rounding up operation $\lceil \bullet \rceil$ is advised at this point. Nevertheless, we decided to implement equation (7.2) in the above shape.

set in any case. For instance, `num_slices_per_part` takes the value 60 (it slightly differs from the `nec_slice_count` because of the addressed roundings). Now, the user can assign the so-called overlap divisor `overlap_div` $\in \{2, 3, \dots, \text{num_slices_per_part} - 1\}$ to define the absolute overlap value as

$$\text{overlap} = \left\lceil \frac{\text{num_slices_per_part}}{\text{overlap_div}} \right\rceil. \quad (7.4)$$

Let's assume we set `overlap_div` = 5 for the example from above, then the overlap is 12 slices thick according to equation (7.4). Because neighboring partitions overlap by this amount, of course the total number of partitions increases, like the subsequent relation

$$\text{partitions} = \left\lceil \frac{N_z}{\text{num_slices_per_part} - \text{overlap}} \right\rceil \quad (7.5)$$

specifies. Continuing the exemplary calculations, the number of partitions mounts to 32 for `overlap_div` = 5.

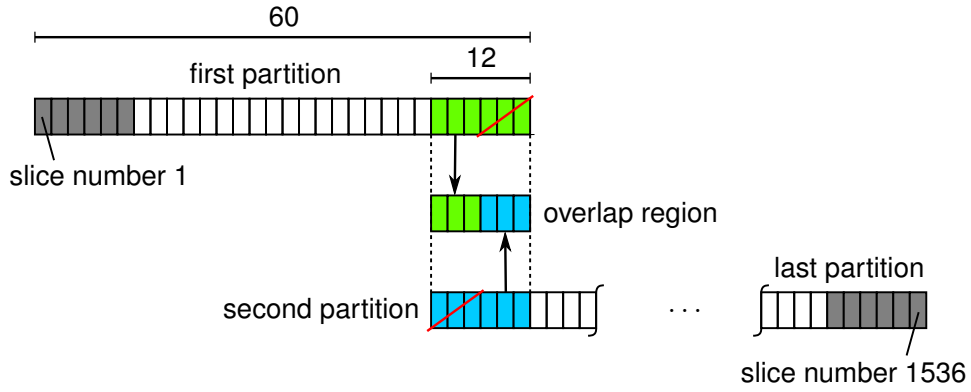


Figure 7.2.: symbolic explanation of the algorithmic treatment of overlap regions

After the overlap has been established, the question arises how the reconstructed slices lying in the overlap region should be determined starting from the existing reconstructions of the two neighboring partitions involved. Certainly there are a few methods to accomplish good images for these slices cleverly, but since this issue is just the last piece of the overall puzzle, we decided to use the simple solution illustrated in Figure 7.2: Each of the two partitions corresponding to the current overlap region writes merely half of the reconstruction data inside the overlap, and in doing so each one uses the slices farther from the border of the individual partition. Accordingly, the canceled slices in Figure 7.2 are not taken into account, which is usually reasonable because of the big gray value fluctuations contained. For `overlap` = 12 in the example, this implies that the last respectively first 6 slices of the two different partitions are omitted. If the slice number of the `overlap` variable is odd, the posterior partition (the one in light blue in Figure 7.2) was appropriated to write one more reconstructed slice. All in all, a good choice of the `overlap_div` can enable us to reconstruct a full experimental cone beam data set with constant quality, except for the global boundary areas (because of the lack of a possible overlap, symbolized by the grey slices in Figure 7.2).

In order to analyze what accounts for a good choice of the overlap divisor and what possible dependencies should be kept at the back of one's mind when applying the proposed method in combination with the ASTRA toolbox reconstruction algorithms, an extensive study has been conducted. The object of investigation is the already known stepped brass cylinder phantom from chapter 6, recorded in the open, modular and flexible in-house XRCT system with the three geometric magnifications $M \in \{4.95, 16.5, 33.02\}$. The reasons for this sample's selection are its plainness and the homogeneous physical properties of the involved material along the object height, implying comparable

gray value distributions for all reconstructed slices (at least within the current step the slice belongs to). Starting from the sinogram data, each data set was reconstructed using the 3D GPU ASTRA algorithm CGLS3D_CUDA (50 iterations), including well-suited definitions of the COR and BHC parameters (compare sections 6.4 and 6.5). Furthermore, the number of slices per partition was kept constant at $\text{num_slices_per_part} = 60$ for comparability reasons, whereas the overlap_div is varied in the set $\{2, 3, 5, 10, 20, 50, \infty\}$. The option “ $\text{ol}\infty$ ” marks the case that each partition attaches oneself to the next one without any overlap. The conclusion of the study has been condensed into the overview plots in Figure 7.3. To get a general idea of the study’s scope, the reader may be referred to appendix A, in which each configuration is depicted separately.

Each diagram in Figure 7.3 is structured as follows: The upper subplot depicts the gray value trend⁴ along the stack height abscissa in blue and the arithmetic mean of the related (e_z -dependent) data by the dashed red line. The dashed green lines give an orientation of the reconstruction quality for the pictured overlap divisor since they demonstrate the standard deviation of the best reconstruction including an overlap (namely $\text{ol}2$), plotted from the mean value in red. The dashed green lines in the lower subplot have the same meaning (and of course also the identical value), but are plotted from the horizontal axis this time. In addition, the lower subplots contain the curves of the difference between the “optimal” reconstruction with $\text{ol}2$ and some selected overlap divisor reconstructions, again compared slicewise along the stack height. The aim of the lower subplot is to justify the decision for a certain overlap_div as a good choice for the belonging geometric magnification.

For $M = 4.95$ in Figure 7.3(a), the key insight that can be obtained is that using **any** overlap that permits omitting at least one slice per partition (which is exemplarily the case for $\text{ol}50$) is clearly advisable, in contrast to no overlap given by the profile indicated by $\text{ol}\infty$. If we compare the gray value trends for $\text{ol}20$ and $\text{ol}50$, the deviation from the best reconstruction with $\text{ol}2$ is insignificant for both of the two parameters. That’s why the upper subplot shows $\text{ol}50$ as the overlap divisor of choice and also reveals the approximate positions where the brass cylinder phantom changes its diameter. Figure 7.3(b) illustrates the situation for the average geometric magnification $M = 16.5$ - herein the periodic outliers coming from the partitioning routine become apparent (their location varies with the overlap divisor, as can be seen with a retrospect to equation (7.5)). An adequate reconstruction quality with respect to the standard deviation of $\text{ol}2$ (green dashed lines) cannot be reached until $\text{ol}5$. Hence, the “optimal” choice concerning computational speed and accuracy at once is achieved with $\text{overlap_div} = 5$. Last but not least, the maximal geometric magnification of $M = 33.02$ included in this study provides the gray value trends in Figure 7.3(c). Although both $\text{ol}5$ and $\text{ol}3$ stay inside the tolerance threshold implied by the standard deviation of $\text{ol}2$, the choice falls on an overlap divisor of 3 with a glance at the upper subplot: Evidently, the oscillations in the gray value trend induced by partitioning cannot be recognized when representing the absolute gray value, whereas they could be identified in such a graph for $\text{ol}5$. Another fact that Figure 7.3(c) adumbrates is how poorly the gray values are for the limits of the stack height (the gray colored slices in Figure 7.2), compare the upper subplot on the left and right margin. These slices are usually taken out of the reconstructed images to that effect by the reconstruction software. As already mentioned, appendix A contains several plotted gray value trends for any overlap scenario addressed above, so for more details please throw a glance at these diagrams.

⁴used pixel regions: rows 310-320, columns 290-300 for $M = 4.95$; rows 900-1000, columns 800-900 for $M = 16.5$; rows 900-1000, columns 700-800 for $M = 33.02$

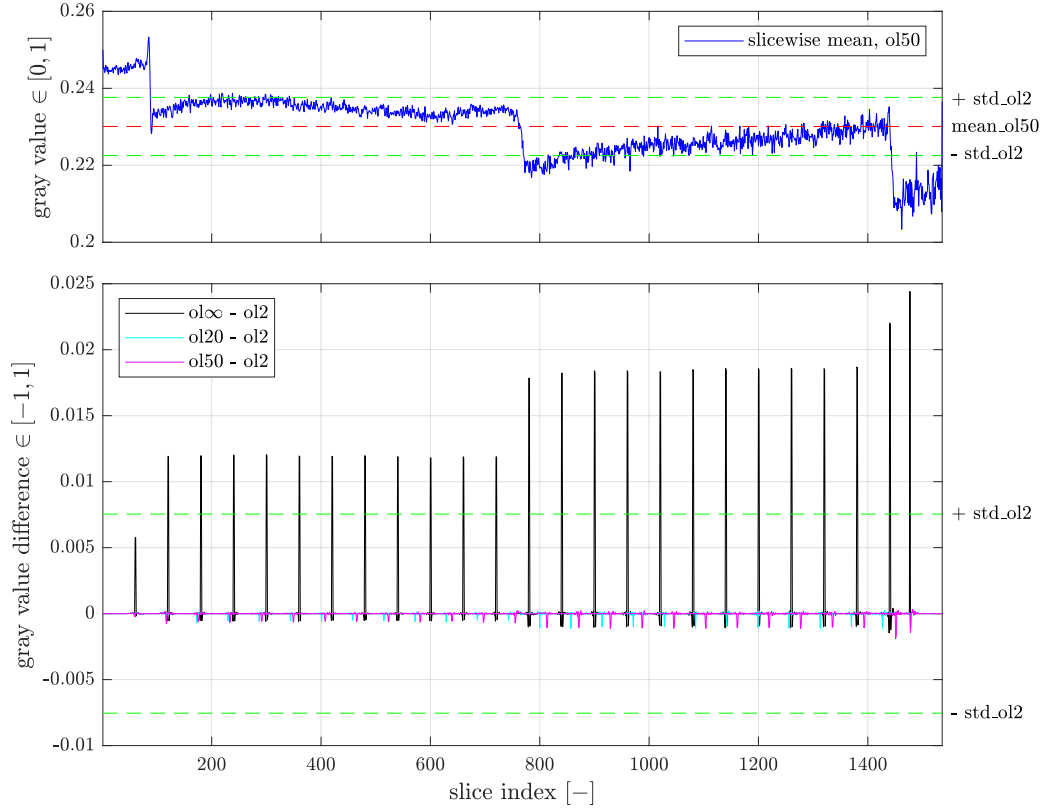
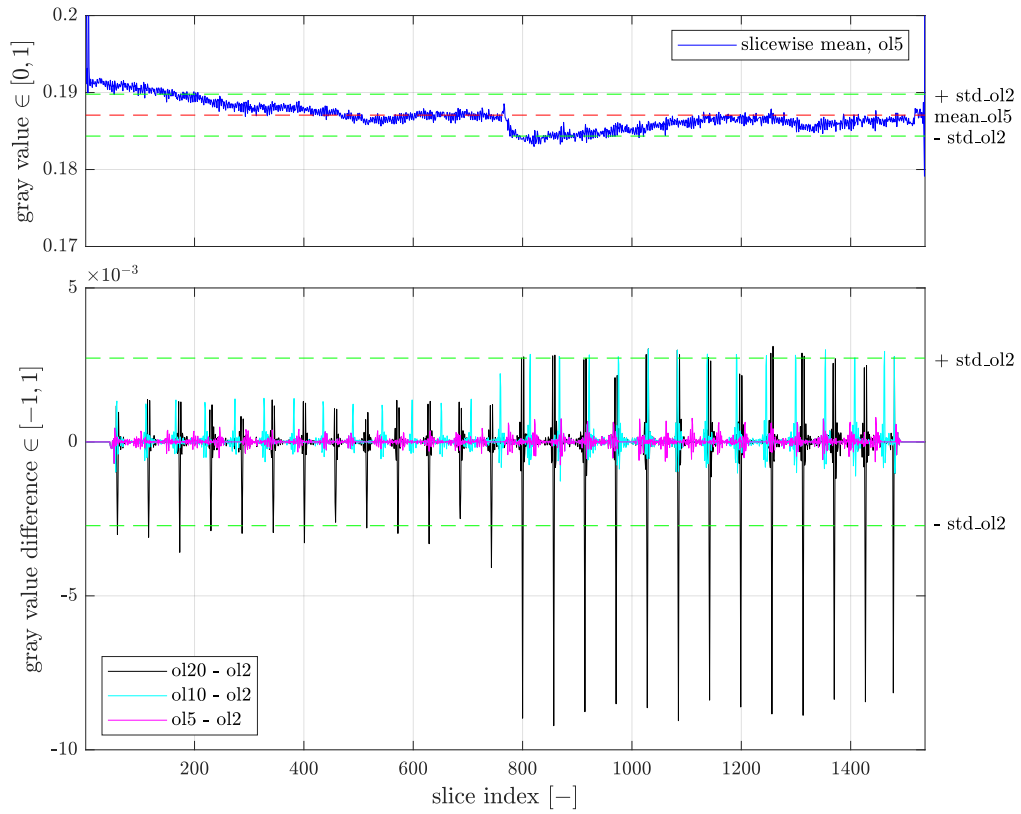
(a) stepped brass cylinder phantom with $M^{(5.1)} \underline{=} 4.95$ (b) stepped brass cylinder phantom with $M^{(5.1)} \underline{=} 16.5$

Figure 7.3.: numerical study of the gray value trends along the slice's stack height for varying geometric magnifications

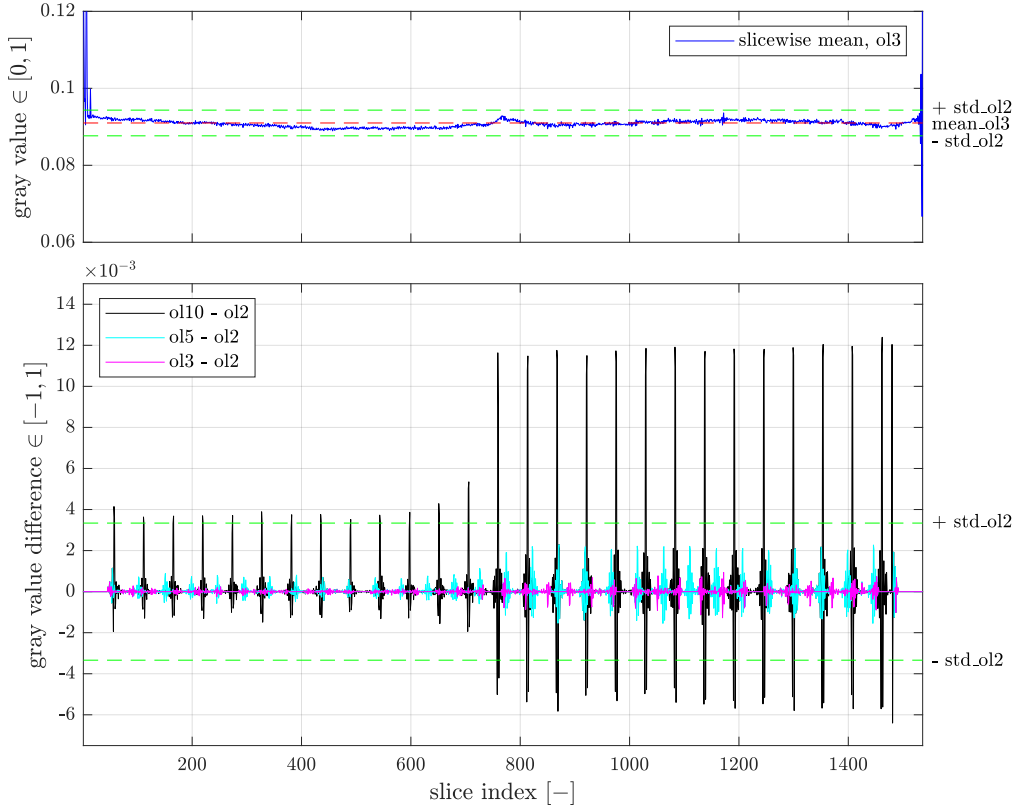
(c) stepped brass cylinder phantom with $M^{(5,1)} \stackrel{=}{=} 33.02$

Figure 7.3.: numerical study of the gray value trends along the slice's stack height for varying geometric magnifications (continued)

overlap_div	2	3	4	5	6	7	8	9	10, 11	12 - 14	15 - 19	20 - 29	30 - 59
overlap (7.4)	30	20	15	12	10	9	8	7	6	5	4	3	2
overwritten ⁵	15	10	7	6	5	4	3			2		1	

Table 7.1.: line-up of slice numbers corresponding to increasing overlap divisors for fixed $\text{num_slices_per_part} = 60$

The conclusions that can be drawn from the stated study are as follows: Firstly, it turned out that the application of any kind of overlap is highly recommended if the user expects comparable gray value ranges along the e_z -direction (which will be the case in general in view of a successive image postprocessing for the reconstructed slices). Furthermore, we found out that it exists a distinct dependency of the essential overlap divisor on the geometric magnification with which the object sample has been projected onto the X-ray detector. Based on the three experimental setups consulted above, Figure 7.4 gives a recommendation by means of an exponential regression which overlap_div should be equipped for which geometric magnification M . The form of the exponential regression curve therein is determined by

$$f(x) = a + b e^{-cx} \quad \text{with} \quad a = 2, b = 62.5 \text{ and } c = 0.1. \quad (7.6)$$

The parameters in equation (7.6) have been fitted with respect to the data points implicated by the

⁵ $\text{overwritten} = \lfloor \text{overlap}/2 \rfloor$

investigated cases for the stepped brass cylinder phantom, but also to two further constraints: For $M \rightarrow 1$ with $M > 1$, we expect that the maximal overlap divisor (which is 59 for `num_slices_per_part` = 60) is assumed. Contrariwise, the minimal overlap divisor 2 makes sense for the highest possible geometric magnifications M (which are only limited by the experimental construction) - inducing a horizontal asymptote by the term $a = 2$. If one used to identify a suited `overlap_div` from Figure 7.4, Table 7.1 directly presents the belonging numeric values for `overlap` and `overwritten` (which stands for the number of slices that will effectively be ignored per partition in the implementation). Table 7.1 also serves to give the reader a deeper understanding for the absolute numbers linked to the described overlap divisors.

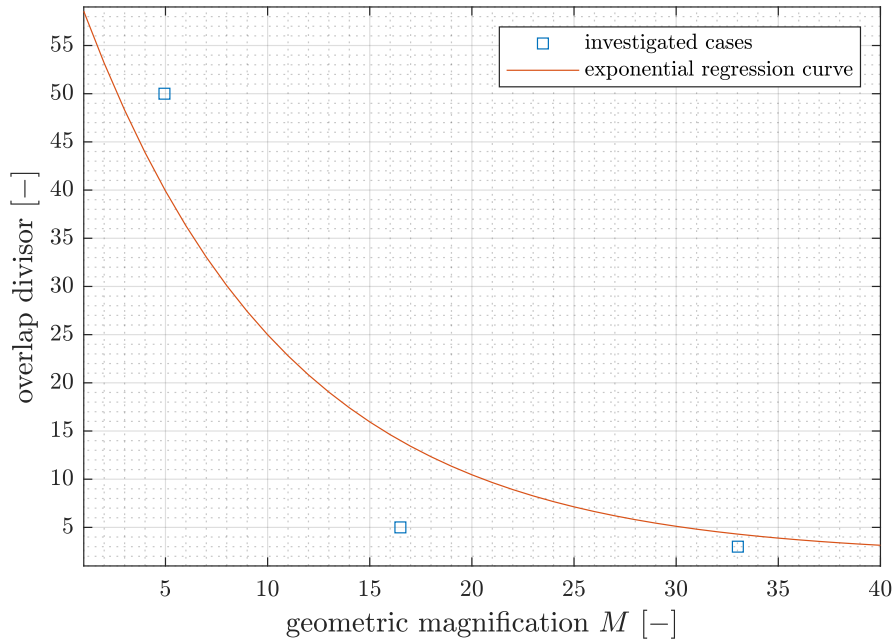


Figure 7.4.: correlation between geometric magnification and required overlap divisor for a proper reconstruction (using 3D ASTRA toolbox algorithms)

7.1.2. Remarks on occurred differences between 2D and 3D reconstruction

In the context of the overlap investigation in the previous subsection 7.1.1, each of the three data sets (for the different geometric magnifications) was also reconstructed with the analogous 2D GPU ASTRA algorithm, in particular `CGLS_CUDA` with 50 iterations, since we could expect that each sinogram contains all required information for one to-be-reconstructed slice in 2D. The COR and BHC parameters remained unmodified compared to the 3D reconstruction including certain overlap divisors. The point of interest by a comparison of these two types of reconstruction algorithms is if the whole effort to achieve a good reconstruction because of the hardware limitations described is worth it. On the one hand, it is mathematically incorrect to try to reconstruct the data sets recorded with cone beam scanning using 2D reconstruction routines. But on the other hand, we can only include a limited number of slices per partition in 3D reconstruction, which is not congruent with the expected slice count `nec_slice_count` from equation (7.1) in our case. Another great motivation for such a contrast is to validate if the prediction of the COR and BHC parameters for 3D made with single slices with 2D reconstruction algorithms is justifiable. In this spirit, a brief comparison seems reasonable - and this will be carried out for the stepped brass cylinder phantom example with $M = 16.5$ in the next paragraph.

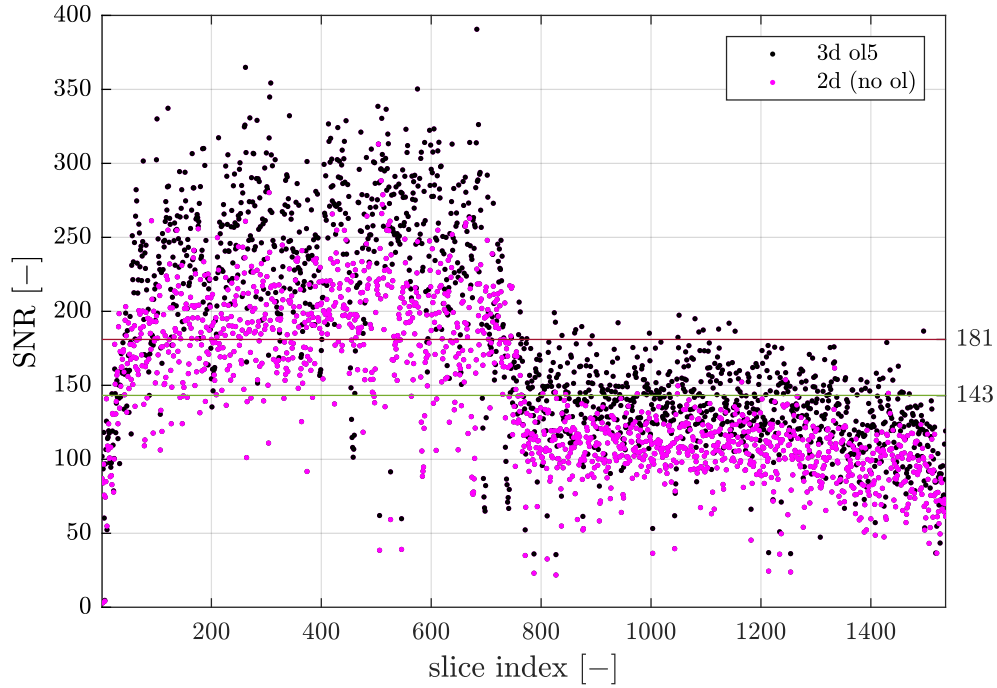


Figure 7.5.: numerical study of the signal-to-noise ratio SNR for the stepped brass cylinder phantom with $M^{(5,1)} = 16.5$

First of all, one should take a look at the running times for the reconstruction of the full data set: While the 3D ansatz with `overlap_div = 5` took about 11600 s (≈ 193 min), 2D reconstruction lasted approximately 16300 s (≈ 272 min), which corresponds to degradation of a factor 1.41⁶. Another observation applies to the gray value trends (cf. Figure 7.3). At large, the absolute gray values of the 2D reconstruction (which are mainly dictated by the toolbox routines) are lower than the ones for the 3D case, whereas the curves show identical trends - for instance, regard the graphs A.2(e) and A.2(h) in appendix A for this. However, it is not obvious which resulting images have better quality characteristics, so that another measure shall be introduced for this contemplation. It is a matter of the so-called signal-to-noise ratio SNR (see “DIN EN ISO 15708-3:2019-09, Zerstörungsfreie Prüfung – Durchstrahlungsverfahren für Computertomographie – Teil 3: Durchführung und Auswertung (ISO 15708-3:2017); Deutsche Fassung EN ISO 15708-3:2019”, 2019), defined as

$$\text{SNR} = \frac{\bar{\mu}}{\sigma} \quad (7.7)$$

with a problem-dependent arithmetic mean $\bar{\mu}$ and the associated standard deviation σ . Altogether, a high value of the SNR is associated with a good reconstruction quality. To contrast the 2D reconstruction of the stepped brass cylinder phantom with $M = 16.5$ with the best choice of a 3D reconstruction according to subsection 7.1.1 (which is ol5 for this geometric magnification), Figure 7.5 depicts the slicewise SNR (which means both the arithmetic mean $\bar{\mu}$ and the standard deviation σ in equation (7.7) refer to a fixed pixel domain⁷ in a single reconstructed slice) for these two scenarios. With just a few downward outliers, both stacks of images show a good reconstruction quality, as measured by the high values of the SNR. If we take the average along the stack height for both scatter plots (compare the dark red and khaki lines in Figure 7.5), it turns out that the mean SNR for the 3D reconstruction with `overlap_div = 5` is almost 40 points higher than the 2D value. Based on

⁶specification of the XRCT lab computer: operating system Windows 7 Professional 64 bit, Intel Xeon E5-2630 v2 processor, 128 GB CPU RAM, NVIDIA GeForce GTX 1080 Ti graphic board, 11 GB GPU RAM

⁷used pixel regions just as in subsection 7.1.1: rows 900-1000, columns 800-900 for $M = 16.5$

the two points carved out, we can record the perception that reconstructing a full cone beam scanning data set with the described 3D procedure from subsection 7.1.1 is advantageous compared to 2D reconstruction if the overlap divisor was assigned well for the specific geometric magnification. Of course we only examined few aspects of a possible extensive comparison, and so no general recommendation should be pronounced because the circumstances are rather problem-dependent.

7.1.3. Application to an irregular real-world data set

In a final step, we will discuss to what extent our advanced algorithmic workaround for the ASTRA toolbox can achieve presentable reconstruction results for an irregular real-world data set. The data set in the focus of this subsection contains the cone beam scanning projections of the open-pored asphalt concrete sample which was already mentioned in Figures 6.2(a) and 6.3. The appendant raw data (Ruf and Steeb, 2020b) can be found in DaRUS (the Data Repository of the University of Stuttgart)⁸. It is composed of $N_\theta = 1440$ single projection images of dimensions $N_z = 1536 \times N_x = 1944$, recorded with a geometric magnification of $M = 2.99$, and also includes one dark and open beam image each. We tried to optimize the reconstructions by a correction of the horizontal COR shift according to subsection 6.4.2 with parameter $\delta u = -2.125$ mm and a BHC as in section 6.5 with coefficients $\{d = 2.5, e = 0.5\}$ for the correction formula in equation (6.6). By default, a ROI mask with ratio 95 % (cf. section 6.3) was used, but no COR tilt correction as described in subsection 6.4.3. Finally, all simulations were run with the ASTRA GPU reconstruction algorithm CGLS3D_CUDA for 50 iterations and a necessary slice count (see the introduction of this section, e.g. equation (7.1)) of 57 slices.

If we look back on the exponential regression curve in Figure 7.4 visualizing the suggested overlap divisor depending on the geometric magnification, the graph prescribes $\text{overlap_div} \approx 50$ for the treated asphalt data set. It should turn out that this selection was estimated much too high since the resulting image stack reveals large fluctuations at the boundary points of two neighboring partitions. While an overlap divisor of 10 was still unable to achieve a good global reconstruction, $\text{overlap_div} = 2$ could finally produce a satisfying outcome, as visible in Figure 7.7. These two mentioned variants for the choice of the overlap divisor will briefly be compared in the following paragraph.

Concerning the running times for the whole reconstruction process, of course ol10 is at an advantage towards ol2 because of the lower number of overlapping slices (and the associated lower partition count), compare Table 7.1: While ol2 requires approximately 17900 s (≈ 298 min), the reconstruction trial for ol10 is finished after about 11300 s (≈ 188 min) - what corresponds to a speeding up of circa 37 %. However, this is at the expense of the reconstruction quality of the obtained images, which can be identified by means of Figure 7.6. It shows graphs of the slicewise mean for the addressed overlap divisors ol10 and ol2 plotted over several slices of the full stack where one transition from one partition to another one is located. This transition lies between the slice indices 705 and 706 and is indicated by the purple vertical line in Figure 7.6. Clearly recognizable are the oscillations of the relative gray value (with respect to the arithmetic mean of the whole trend for the displayed slice indices) with an amount of - 10 % through + 35 % in proximity to the partition border for the red curve symbolizing ol10, quite contrary to the almost flat curve for ol2 in blue. Furthermore, the vertical lines in dark green and yellow represent the borders of the overlap area of ol10 respectively ol2 for the two associated partitions in the background. These additional information helps us to make the statement that the number of overwritten slices (cf. Table 7.1) is insufficient for $\text{overlap_div} = 10$ as there still occur major gray value outliers for the five slice indices close to the partition transition on each side. The closing realization from Figure 7.6 is that the overlap

⁸<https://doi.org/10.18419/darus-639>

divisor ol2 can generate a consistent stack of reconstructed slices (disregarding the global borders of the stack) for the specified irregular asphalt sample data set, and so the overall aim of the present thesis is qualified as fulfilled. Figure 7.7 gives an impression of the internal structure of the investigated open-pored asphalt concrete sample at different heights inside the stack of reconstructions. In addition, Figure 7.8 shows sectional views through the whole stack of reconstructed images along the horizontal in Figure 7.8(a) respectively the vertical center line of the reconstructed slice each in Figure 7.8(b).

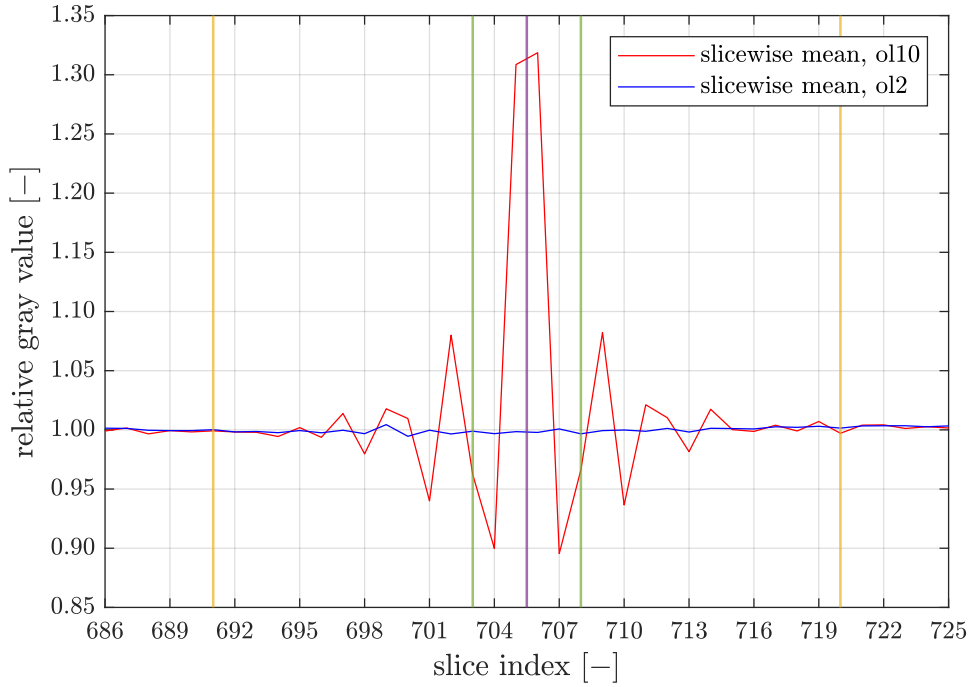
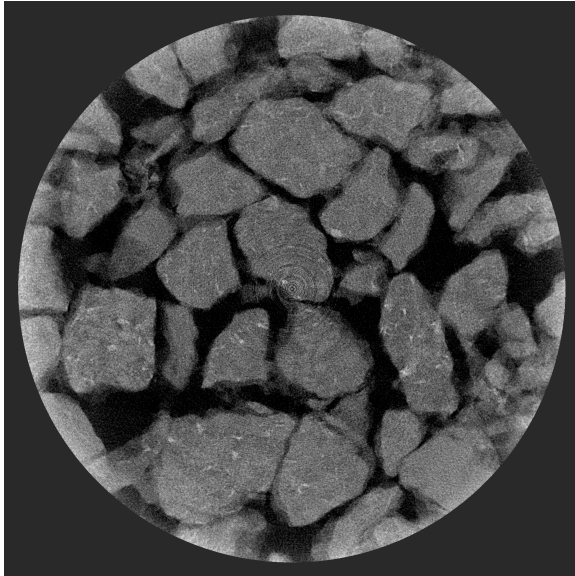
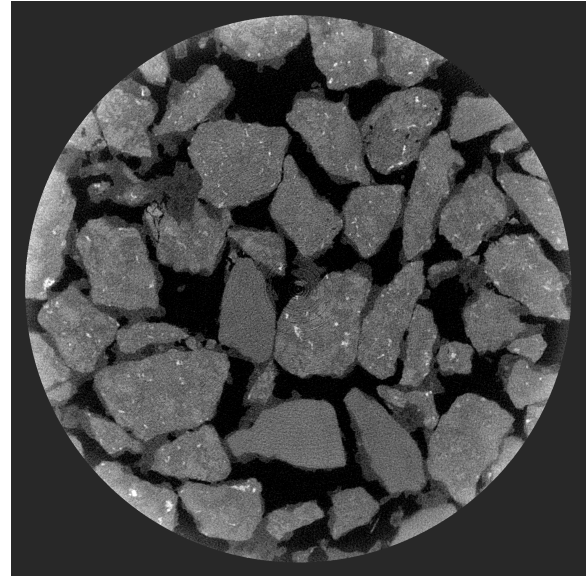


Figure 7.6.: numerical study of the mean gray value trends for selected slices of reconstructions for the open-pored asphalt concrete sample with $M^{(5.1)} \approx 2.99$

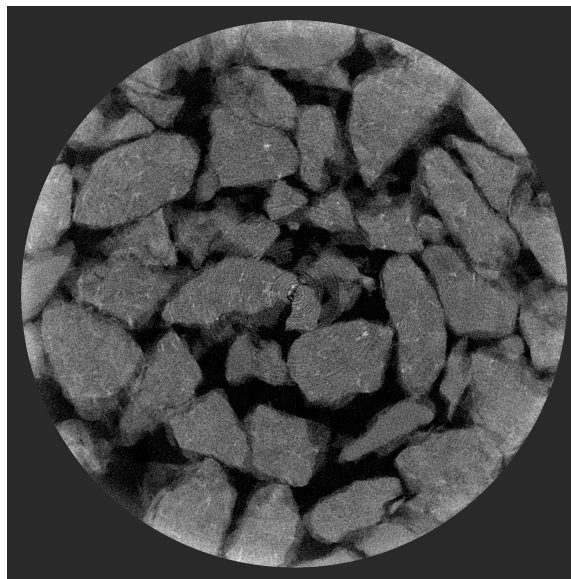
Possible reasons why the predicted `overlap_div` from subsection 7.1.1 is unsatisfactory cannot be found that obvious. One main occasion will definitely be the fact that the irregular real-world asphalt sample implies highly inhomogeneous input projection data and sinograms (for instance, see Figure 6.2(a)), which does not apply in the same way to the stepped brass cylinder phantom made up of homogeneous matter. In addition, the object under investigation captures the whole reconstruction image (after applying the ROI mask) for the asphalt sample because the physical dimension of the material is much larger than the FOV of the experimental scanning setup. For the stepped cylinder phantom with comparable geometric magnification $M = 4.95$, this certainty was not given. That phantom inclusive its surrounding polyester resin layer only covered about 30 % of the diameter pixels and thus approximately 9 % of the total utilizable pixel area for reconstruction. Potentially, the user notices minor variations of the gray value for successive slices better if the reconstructed images show more details and heterogeneities. Nonetheless, the addressed granularity can also be identified as a subordinate merit of 3D reconstruction in contrast to 2D reconstruction, as we could realize for the open-pored asphalt concrete sample used in this subsection.



(a) for slice 384 of 1536

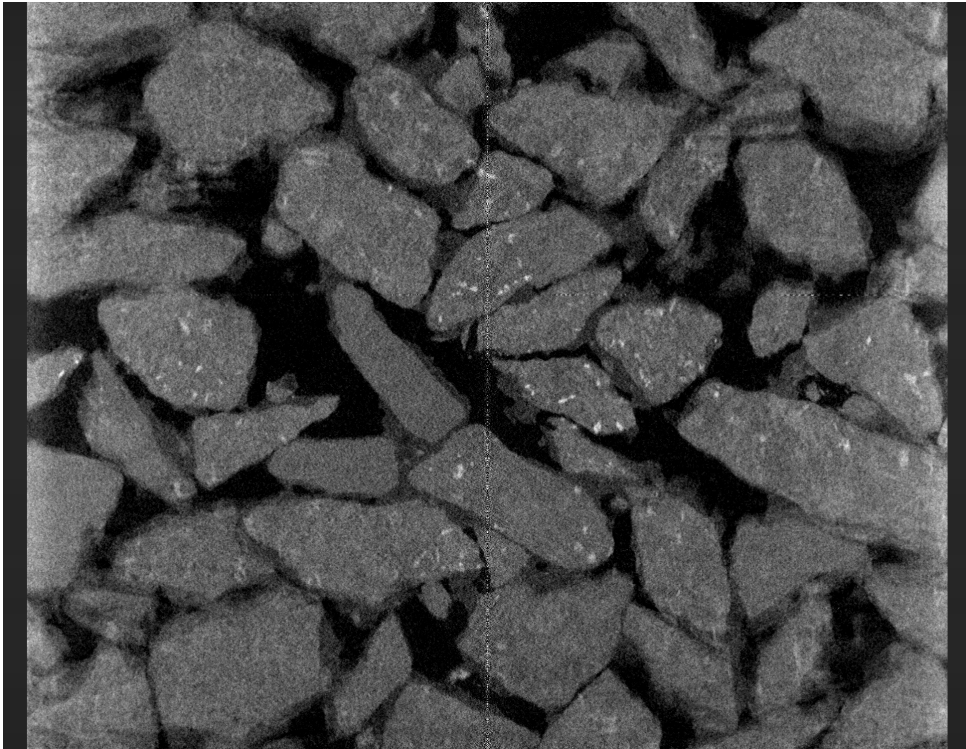


(b) for slice 768 of 1536 (the medium slice)

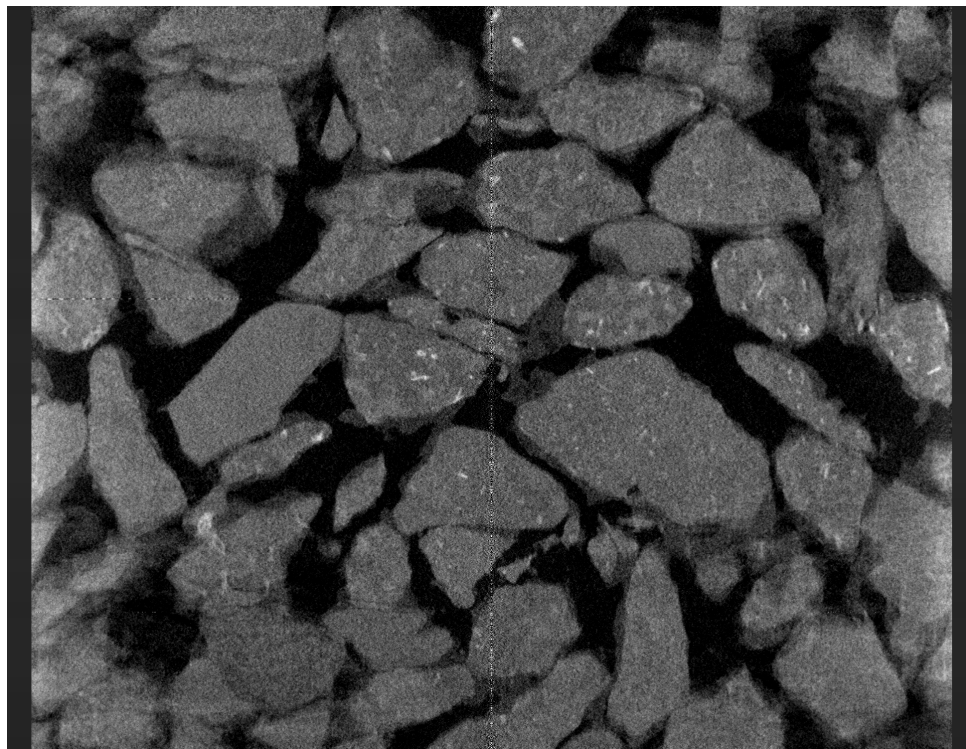


(c) for slice 1152 of 1536

Figure 7.7.: reconstructed images of the open-pored asphalt concrete sample
with $\text{overlap_div} = 2$



(a) for the horizontal center line at row 972 of 1944



(b) for the vertical center line at column 972 of 1944

Figure 7.8.: sectional views of the open-pored asphalt concrete sample reconstructed with $\text{overlap_div} = 2$ along the e_z -direction (each image row pictures the gray values along a specified line in one reconstructed slice, for example in the ones given in Figure 7.7)

7.2. Treatment of helical cone beam scanning data

The approach to reconstruction of data sets recorded with helical cone beam scanning considerably differs from the one for cone beam scanning in the last section 7.1. One can realize this by a consideration of the following example: Let's assume we only have one single sinogram of a helical cone beam scanning setup like the one pictured in Figure 6.2(b) at our disposal. Correspondingly, the virtually depicted detector is composed of the exclusive middle detector row, implying $\text{num_sino_used} = 1$ (even if the physical detector would offer more available detector rows, $N_z > 1$). Nevertheless, we can try to start a 2D-like reconstruction with this one sinogram, as it contains information about all the voxels included through the vertical translation of the measured matter. The exact relation can be found as

$$\text{num_rec_slices} = \left\lceil \frac{z_dist}{\tilde{p}_z} \right\rceil + \text{num_sino_used} \quad (7.8)$$

by geometrical considerations for the case of known right-hand-side parameters. For instance, the one sinogram in Figure 6.2(b) yields $\text{num_rec_slices} = 2401^9$, which is of course mostly dictated by the first term in equation (7.8).

If we want to include more than only one sinogram ($1 < \text{num_sino_used} \leq N_z$), the position of the associated detector rows relative to the central row of the physical detector gains importance. Clearly the number of voxels increases for a higher number of used sinograms according to equation (7.8), but the crucial point is that the trans_vec of the vectorized helical cone beam projection geometry (cf. section 5.2) needs to be modified. The reason for this is that the ASTRA volume geometry creates a cuboid (with dimensions $N_x \times N_x \times \text{num_rec_slices}$) in whose center the overall coordinate system is fixed, and also the alignment of the projection geometry has to be oriented relative to this origin.

For this purpose, we have to differentiate between three dissimilar arrangements of the involved detector rows with respect to the central physical detector row. An additional distinction has to be made regarding the number of detector rows N_z , but this just induces slight adaptations for one of the foregoing cases. To support the comprehension, all possible variants shall be sketched in Figure 7.9 for two simplified examples, in which the main difference consists in the even respectively odd number of detector rows. The first constellation is highlighted in orange and shows the case that the entire considered detector rows lie above the central detector row (since both $\text{start_idx} = 1$ ($2 \leq \text{mid_idx}^{10}$ and $\text{end_idx} = 3 \leq \text{mid_idx}$). Here, the projection geometry is to be shifted in positive e_z -direction by

$$\text{upper_extra} = \frac{\text{end_idx} - \text{start_idx} + 1}{2} = \frac{\text{num_sino_used}}{2}, \quad (7.9)$$

which results in $\text{upper_extra} = 1.5$ (1). Quite similar is the situation colored in lime green where every processed detector row is below the central detector row ($\text{start_idx} = 8 > \text{mid_idx}$; $\text{end_idx} = 9 > \text{mid_idx}$ for the even number of detector rows on the left or $\text{start_idx} = 9 > \text{mid_idx} + 1$; $\text{end_idx} = 9 > \text{mid_idx} + 1$ for the odd number of detector rows on the right). This time, the prepared ASTRA projection geometry has to be displaced by the value

$$\text{lower_extra} = \frac{\text{end_idx} - \text{start_idx} + 1}{2} = \frac{\text{num_sino_used}}{2} \quad (7.10)$$

in negative e_z -direction, yielding $\text{lower_extra} = 1$ (1.5) for the two instances in Figure 7.9. The third arrangement for the case of an **even** number of detector rows N_z is defined exactly by the

⁹used parameters of the scanning setup: $z_dist = 30$ mm, $\tilde{p}_z \approx 12.5003$ μm

¹⁰ $\text{mid_idx} = \lfloor N_z/2 \rfloor$

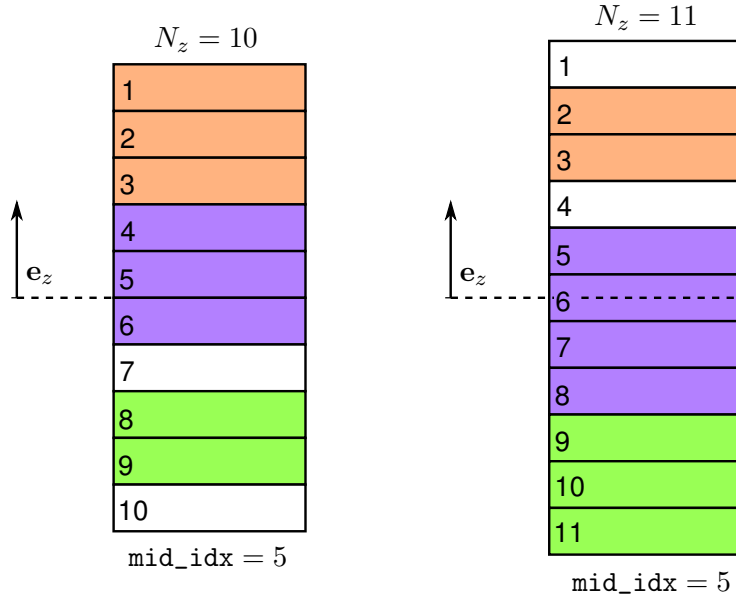


Figure 7.9.: schematic, exemplary representation of all imaginable combinations that affect the orientation of the ASTRA projection geometry for helical cone beam scanning

conditions $start_idx \leq mid_idx$ and $end_idx > mid_idx$. For the example illustrated on the left side in purple with even $N_z = 10$, this holds true as $start_idx = 4 \leq mid_idx$ and $end_idx = 6 > mid_idx$. To achieve the expected shift of the projection geometry by $+0.5$ along the e_z -axis, the two sizes from equations (7.9) and (7.10) are redefined via

$$upper_extra = \frac{mid_idx - start_idx + 1}{2}; \quad lower_extra = \frac{end_idx - mid_idx}{2}. \quad (7.11)$$

The superposed total shift is obviously given by

$$total_extra = upper_extra - lower_extra \quad (7.12)$$

and so the desired value $total_extra = +0.5$ can be obtained. As one can imagine well, the combination of an even number of detector rows N_z and an even number of used sinograms num_sino_used , arranged symmetrical to the central detector row, is quite familiar and luckily results in $total_extra = 0$ under these circumstances. The last possibility is displayed in purple on the right side of Figure 7.9, characterized by the **odd** number of detector rows $N_z = 11$ and the lightly modified constraints $start_idx \leq mid_idx + 1$ and $end_idx \geq mid_idx + 1$. Both of the conditions are fulfilled for the example given since $start_idx = 5$ and $end_idx = 8$ (remember the definition $mid_idx = \lfloor N_z/2 \rfloor$ from above). For an odd number of detector rows, equation (7.11) needs to be adjusted to

$$upper_extra = \frac{mid_idx - start_idx + 1}{2}; \quad lower_extra = \frac{end_idx - mid_idx - 1}{2}. \quad (7.13)$$

The superposition to $total_extra$ still follows the instruction from equation (7.12), and hence $total_extra = -0.5$ results for the pictured example.

The detailed explanation of the exact alignment of the projection geometry in the last paragraph might seem overdone to the reader for a testing problem like the one portrayed in Figure 7.9, but a precise readjustment with the calculated `total_extra` translation is immensely important for real-world scan data. To implement this displacement, the last code snippet of section 5.2 should be altered to:¹¹

```
% adjustment of the vectorized projection geometry to 3D helical
% cone beam using the vector trans_vec
trans_vec = z_trans.*(0:(num_angles - 1))' - z_dist/2 - total_extra;
proj_geom.Vectors(:, 3) = proj_geom.Vectors(:, 3) + trans_vec;
proj_geom.Vectors(:, 6) = proj_geom.Vectors(:, 6) + trans_vec;
```

After we studied the accurate interaction of the necessary ASTRA volume and projection geometry, the next level problem was to reconstruct one sinogram like the one already mentioned several times in Figure 6.2(b). But again, the amount of data upset our plans: This sole sinogram with the dimensions $N_\theta = 2880 \times N_x = 830$ would of course fit into the GPU RAM (memory requirements about 19.12MB), but the generated volume geometry necessitates a size of $N_x = 830 \times N_x = 830 \times \text{num_rec_slices} = 2401$, which corresponds to a more than seven times higher GPU RAM consumption than we used to identify as available in section 7.1. The principal reason for the high number of voxels in e_z -direction consists in the big value for the total traveled z -distance `z_dist`, which induces the size `num_rec_slices` according to equation (7.8). At this point, one could of course wonder about possible loopholes to overcome this repeated sticking point (like a partitioning of the sinogram rows and the associated voxels together with an additive management of the reconstruction output data to approximate the global theoretical gray values for the considered slices), but the finite time horizon of this master thesis left this job site open. Instead, we will present a theoretically performed test example in the successive closing paragraph of this chapter to demonstrate that our implementation around the ASTRA toolbox can in principle map the reconstruction of helical cone beam scanning data.

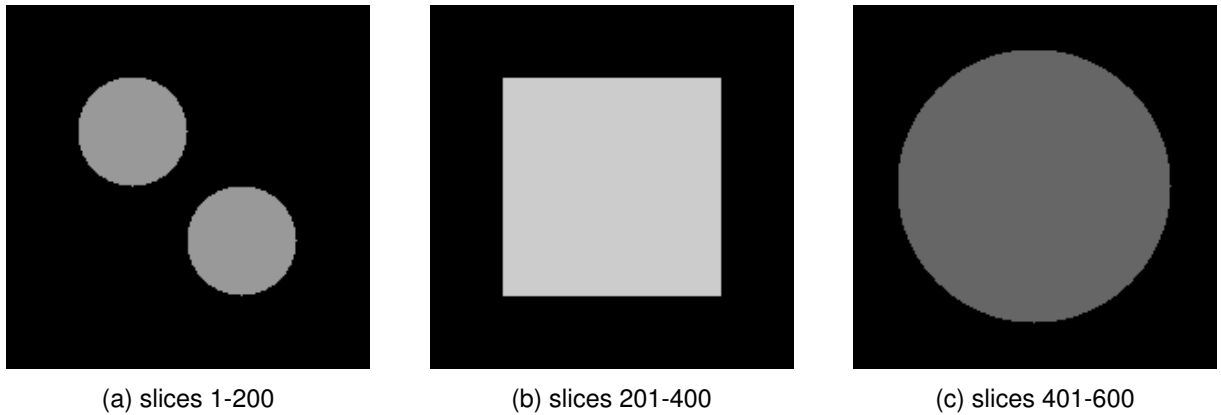


Figure 7.10.: visualization of the underlying exact voxel data in different heights
(counted from above)

The test example is constructed as follows: We prescribe an elongated test phantom by a $N_x = 200 \times N_x = 200 \times \text{num_rec_slices} = 600$ voxel cuboid which possesses three equally sized sections along its altitude with variable cross sections, as depicted in Figure 7.10. The first, superior segment (with slice indices 1 to 200) is made up of two eccentric circles with a radius of 30 pixels

¹¹For the arrangements in orange and lime green from Figure 7.9, of course `total_extra = upper_extra` respectively `total_extra = - lower_extra` applies. The total shift has to be subtracted because of the forced sign definition of the ASTRA toolbox, analogous to the sign of the `z_dist`-fraction.

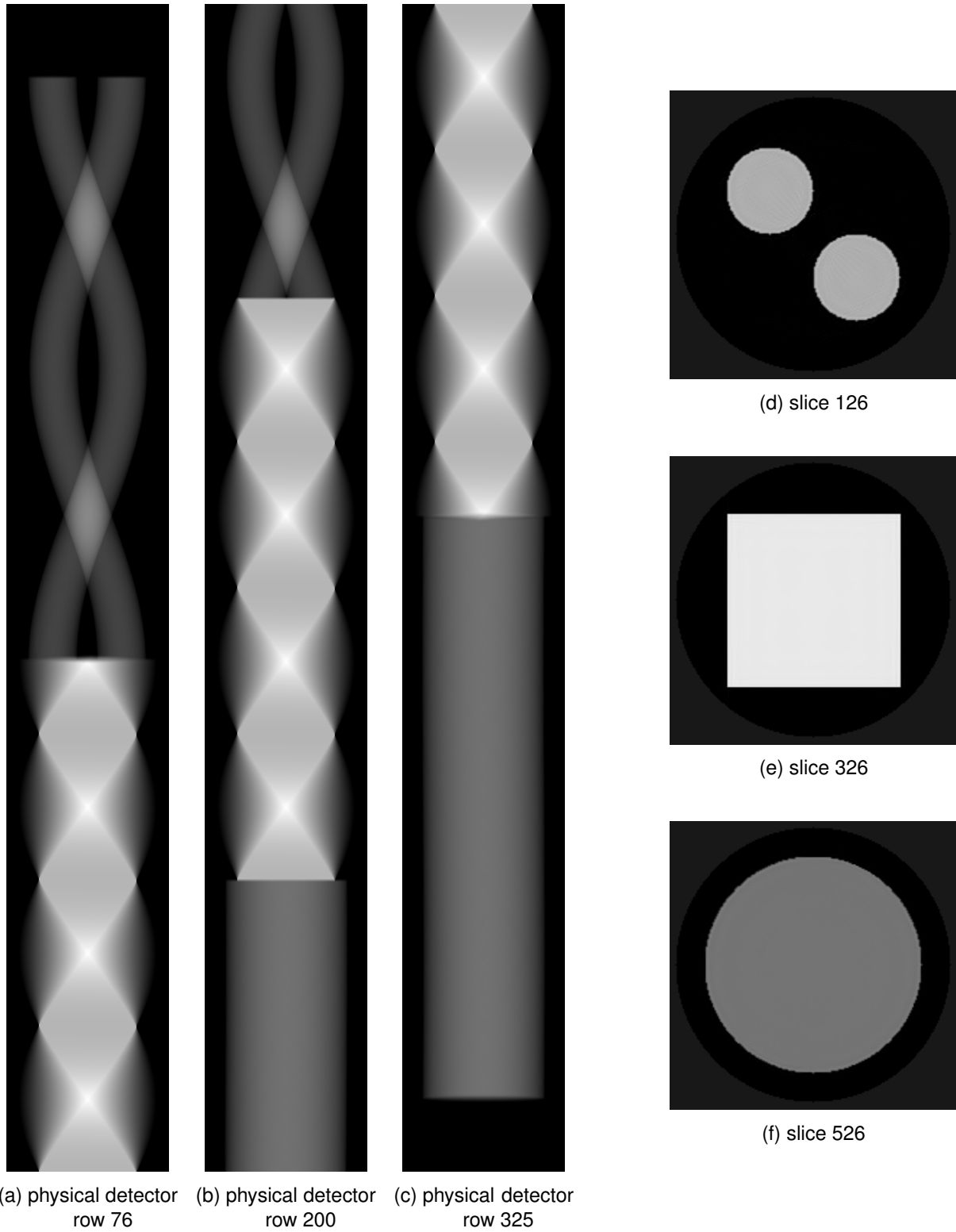


Figure 7.11.: sinograms ((a) - (c)) and reconstructions ((d) - (f)) of the test phantom generated with the specified helical cone beam scanning projection geometry

and a fixed relative gray value of 0.6. This is followed by the central section (slice indices 201 to 400), which includes a square with edge length of 120 pixels and a known relative gray value of 0.8. The slice indices 401 through 600 belong to the final segment and show one centered circle with a 75 pixels radius and an assigned relative gray value of 0.4. To map a realistic empirical helical cone beam scanning setup in the ASTRA projection geometry, we were oriented towards scan parameters which would be consistent with the composition of our XRCT lab. At this point, it is sufficient to present exemplary parameters like the number of detector rows $N_z = 400$, the voxel size $\tilde{p}_z \approx 12.4667 \mu\text{m}$, the number of projection angles $N_\theta = 1440$ along $\text{num_turns} = 2$ entire 360° turns and the total covered z -distance $z_dist = 5 \text{ mm}$ (as before, this z -pitch quantity is measured in the iso-center). The choice of z_dist in interaction with the voxel size \tilde{p}_z causes that we have to select at least $\text{num_sino_used} \geq 198$ (enforced by equation (7.8), resolved after the unknown size) - so we defined $\text{num_sino_used} = 250$ here. All this led to 250 sinograms (which were simulated forward with ASTRA toolbox routines) of dimensions $N_\theta = 1440 \times N_x = 200$, of which the first, middle and last one are pictured in Figure 7.11(a) to 7.11(c). The caption of the images refers to row indices on a possible physical detector with $N_z = 400$ many rows (where the supreme detector row is numbered with 1) and the fact that we use a higher number of num_sino_used as essential is reflected in the blank pixels in the top of Figure 7.11(a) respectively in the bottom of Figure 7.11(c). Finally, we started a reconstruction with the created sinograms, in which the only further optimization applied was the standard ROI mask with ratio 95 % (compare section 6.3). The simulation once again used the ASTRA GPU reconstruction algorithm CGLS3D_CUDA with an iteration count of 50, this time including all the $\text{num_sino_used} = 250$ disposable sinograms at one go. The findings of the reconstruction for the test phantom are quite promising since most of the reconstructed slices faultlessly reflect the initial situation. Figures 7.11(d) through 7.11(f) verify this proposition because they picture the reconstruction images correlated with the exact input voxel data from Figure 7.10 (the slice indexing is again modified because of the definition $\text{num_sino_used} = 250$). Solely the reconstruction quality in the upper and lower border area of the voxel stack is inadequate - which reflects our expectation as the number of available projection information contained in the sinograms decreases in this regions according to the recording technique of helical cone beam scanning. All in all, we can make the statement that all the considerations regarding helical cone beam scanning have been constructive and the transfer to the more demanding experimentally measured data sets should be the next milestone to achieve.

8. Conclusion

The reconstruction of μ XRCT data sets is the essential step from experimentally recorded projection images to interpretable voxel information which try to represent the internal structure of the object under investigation. To obtain an insight into the interior of a sample in a noninvasive and nondestructive way is in great demand in many fields of application, for instance in human medical imaging and materials science. The work at hand provides the opportunity to reconstruct such μ XRCT data sets to a certain degree using the established ASTRA toolbox. The whole requisite previous knowledge concerning X-ray physics, experimental scan setups and reconstruction itself (including different mathematical approaches to this task) are elucidated in chapters 2 to 4. The second half of this thesis (chapters 5 through 7) deals with the description of the ASTRA toolbox features, with some image processing and filtering methods added to the basic ASTRA routines and in a final step with the results of its utilization for experimental full 3D cone or helical cone beam scanning data sets.

Summing up, it can be said that the ASTRA toolbox measures up to our expectation that more than just theoretically respectively mathematically motivated test examples can be reconstructed, but not offhand. Over and above standard preparatory tasks such as normalization, the user has to handle the correct manipulation of the experimental scan parameters. Important techniques to improve the reconstruction image quality for real-world projection data like BHC and COR misalignment correction are absent, and a self-made implementation of them often requires the usage of the vectorized projection geometry representation, which in turn is incompatible with the application of the powerful FBP (in 2D) or FDK (in 3D) reconstruction algorithms. Nevertheless, we could achieve a meaningful reconstruction of an entire experimental cone beam scanning data set, for which all developed methods had to be applied, and even a theoretical test example could be reproduced for the demanding helical cone beam scanning setup.

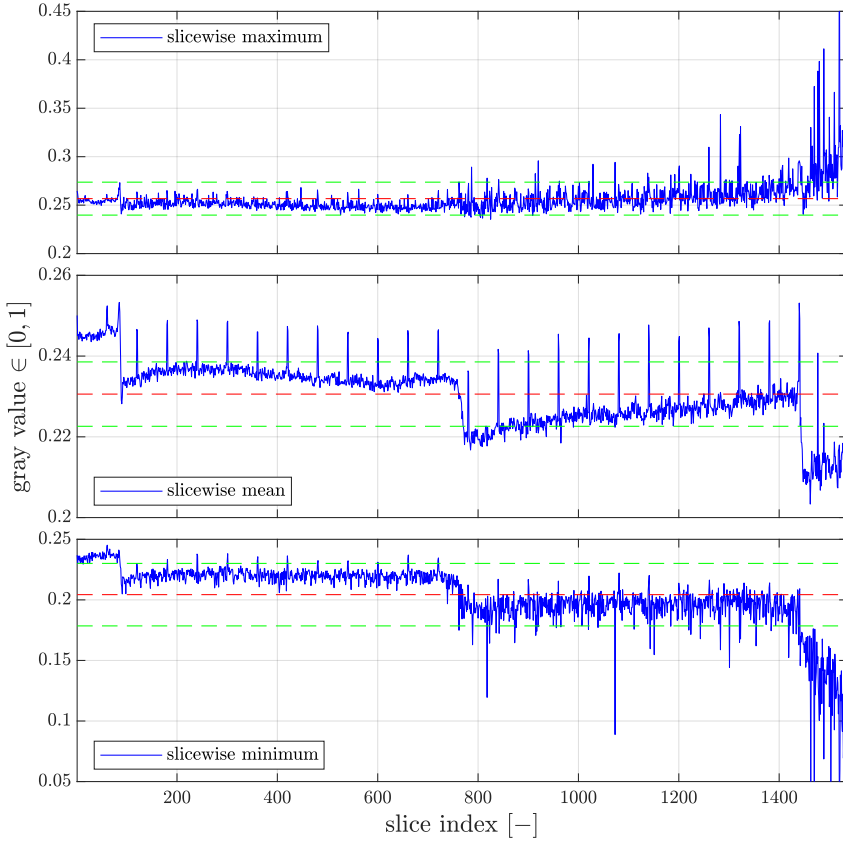
As already discussed, the rework of the irregular real-world cone beam projections was really challenging due to insufficient GPU memory availability (which could be resolved by the introduction of the so-called overlap). Here, the question is raised if or how the developers provided for the treatment of the amount of data incidental with such real-world data sets, especially because a second detector on hand in the in-house XRCT lab with an even higher spatial resolution has not been used for the projections' recording yet. Proceeding research on the very spot can of course deal with a possibly better choice of the gray values for the so far overwritten slices inside the overlap region and a more differentiated study with respect to all potential influencing factors could enrich the presented overlap study. It would be valuable to the same degree to analyze if a similar partitioning approach (at which the segmentation could for instance split the sinogram rows) also yields fruit for experimental helical cone beam scanning projections. In general, one could engage in the problem if and how the employment of multiple graphic cards can contribute to a faster or easier reconstruction with the ASTRA toolbox basic routines. Apart from the challenges linked to hardware limitations, further perspectives to enhance the resulting image quality open up with a deeper familiarization in the suppression of beam hardening and ring artifacts. Perhaps the implementation of a more complex computational BHC method in combination with the introduction of a (sinogram-based) ring artifact reduction lead to a distinct increase of the reconstruction quality even for heterogeneous sample materials. The incorporation of a spot filter for the sinogram images could also offer the

chance to contribute to a strong improvement of the reconstructed images. Finally, a minor further contact point for succeeding work lies in the validation of the theoretically established COR tilt correction so that potentially occurring experimental misalignments of this kind can in future be resolved computationally inside the reconstruction process.

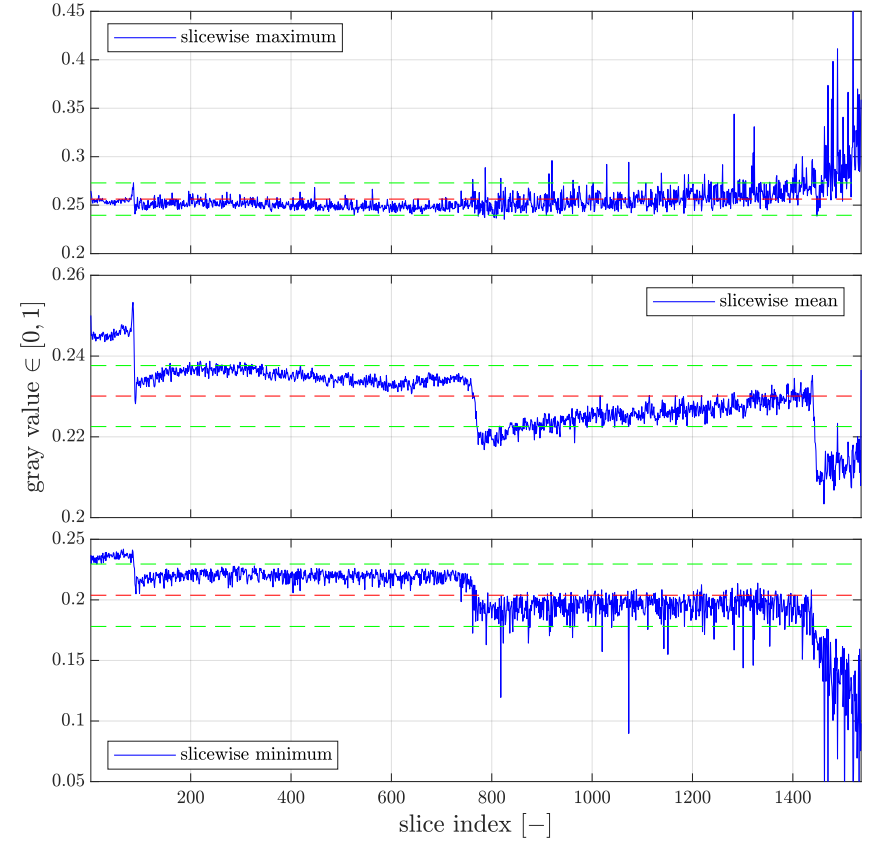
A. Appendix

Complementary plots of the gray value trends for the overlap study in section 7.1.1

Figures A.1 through A.3 are structured in the following way: Each single diagram contains subplots with the gray value trends for the slicewise maximum, arithmetic mean and minimum (starting from the top). The red and green dashed lines of each subplot indicate the arithmetic mean respectively the standard deviation, calculated along the stack height for the pictured gray value data. Furthermore, the ordinates of every subplot for $\text{overlap_div} \in \{2, 3, 5, 10, 20, 50\}$ are identically chosen to ensure comparability in between these diagrams. This could not be achieved for the overlap divisor ol_∞ (too strong oscillations) and the 2D reconstruction (absolute gray values lie below the level of the absolute gray values for the images reconstructed with 3D algorithms).

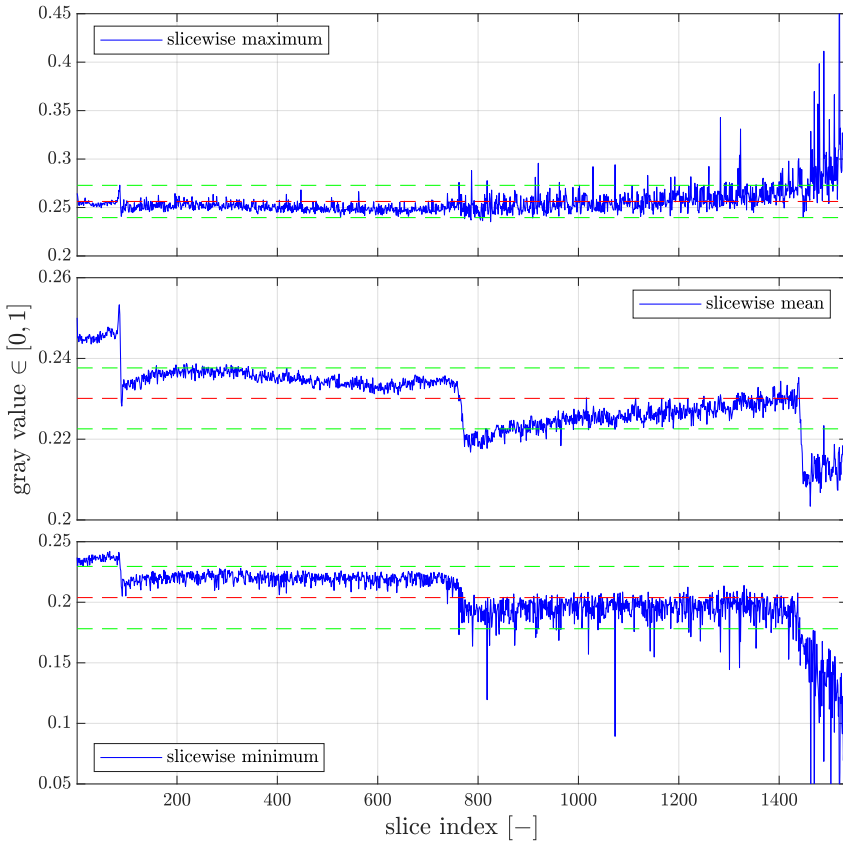


(a) without applying any overlap ("ol ∞ ")

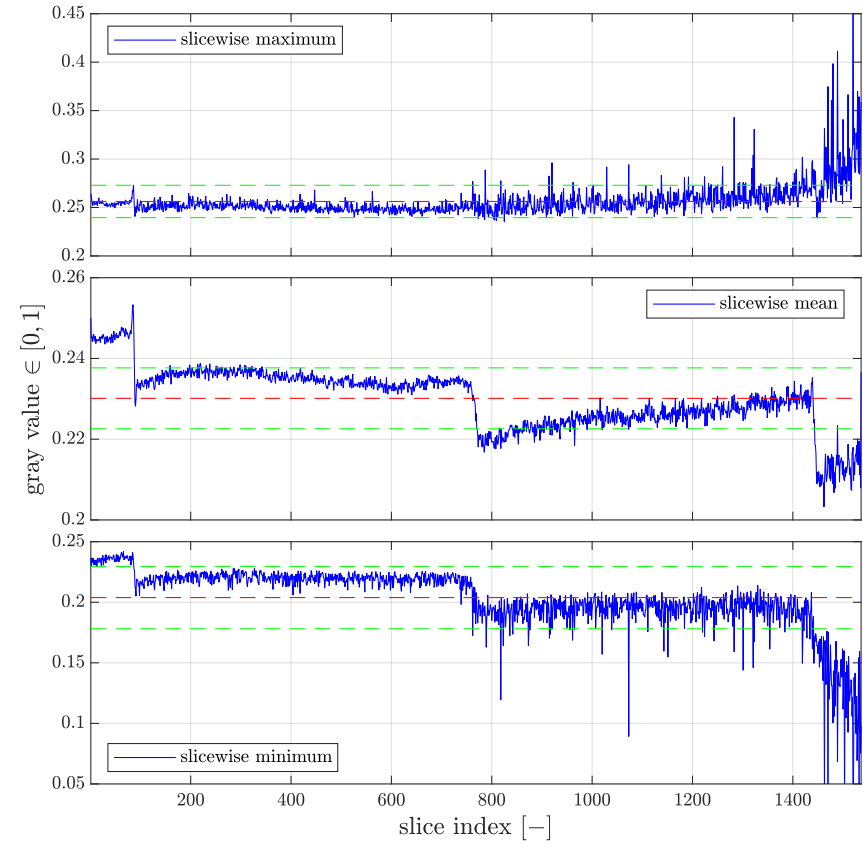


(b) for overlap divisor ol50

Figure A.1.: numerical studies of the gray value trends along the slice's stack height for $M \stackrel{(5.1)}{=} 4.95$

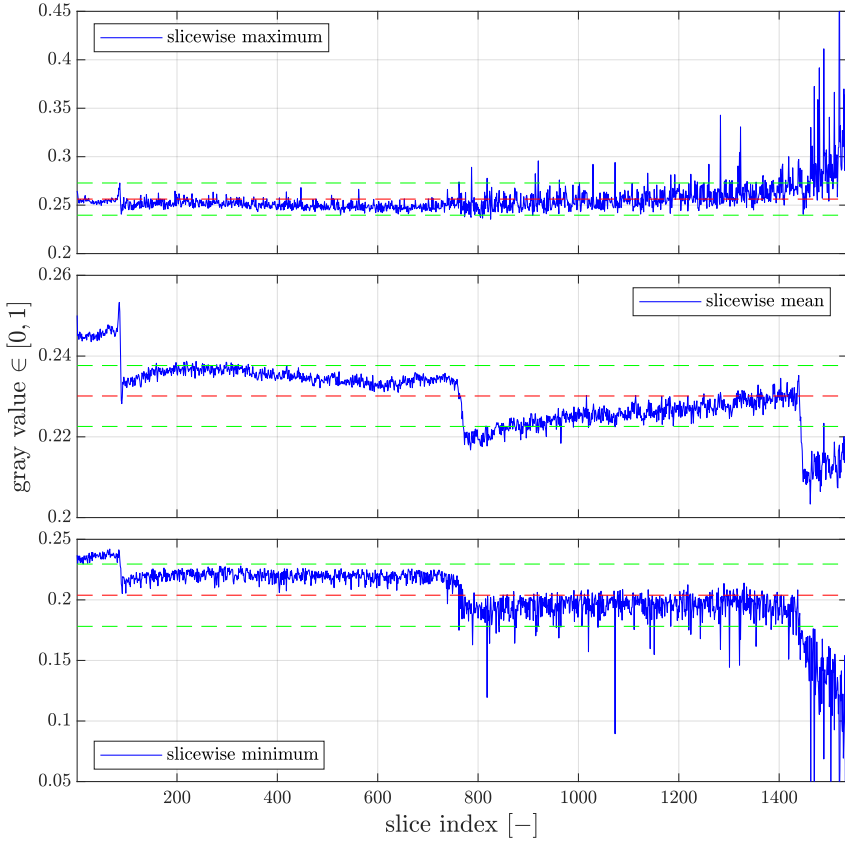


(c) for overlap divisor ol20

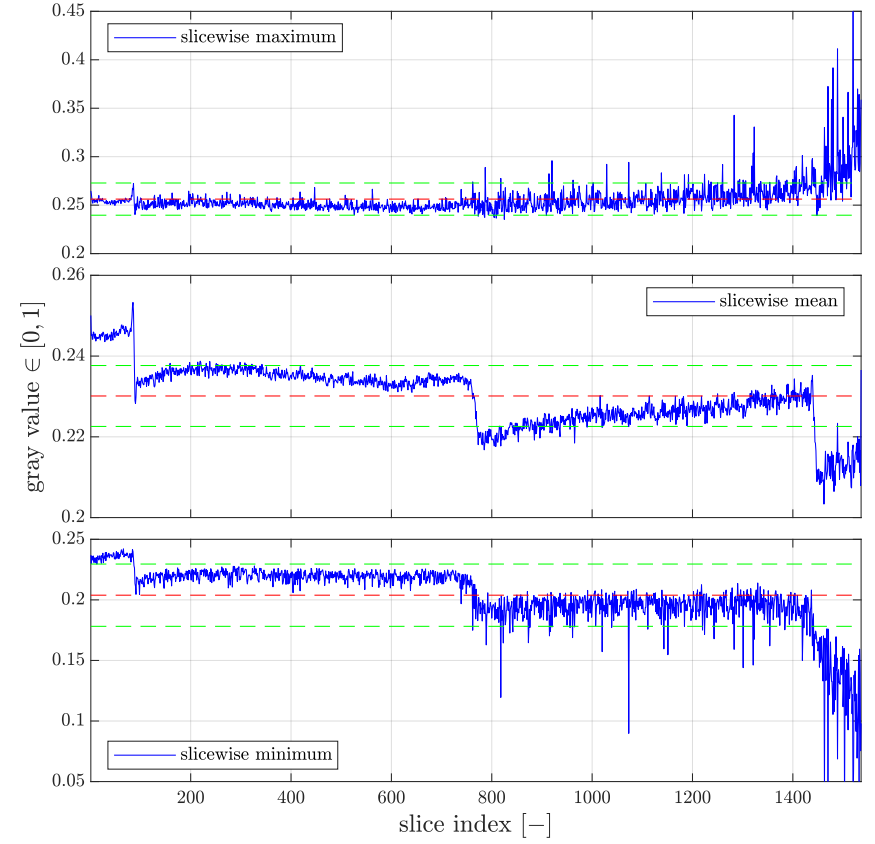


(d) for overlap divisor ol10

Figure A.1.: numerical studies of the gray value trends along the slice's stack height for $M \stackrel{(5.1)}{=} 4.95$ (continued)

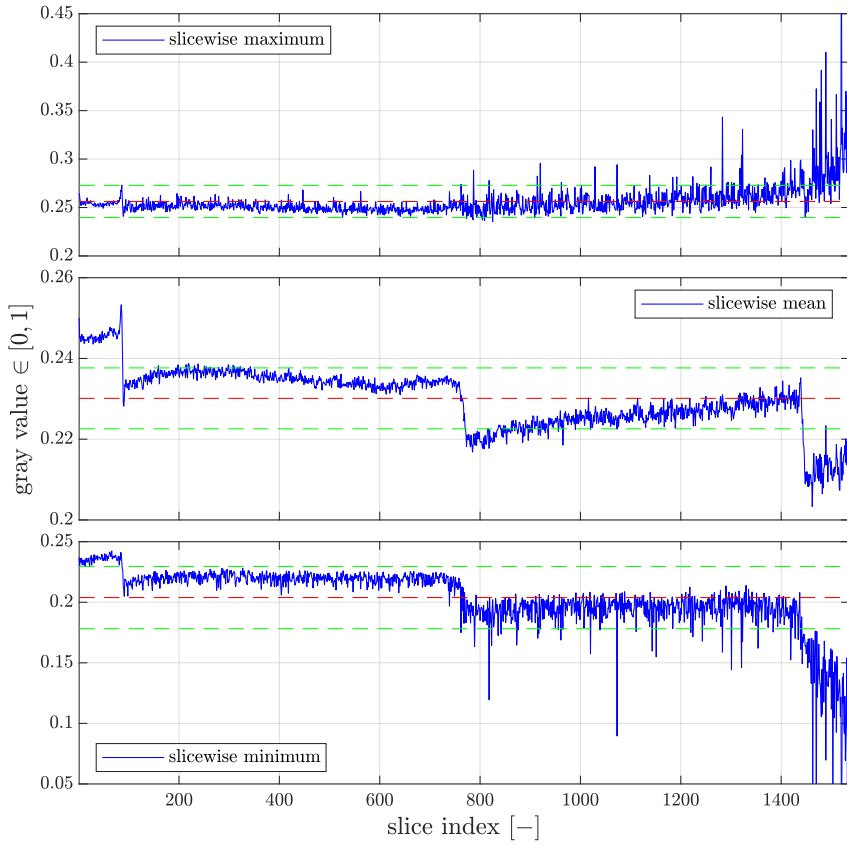


(e) for overlap divisor ol5

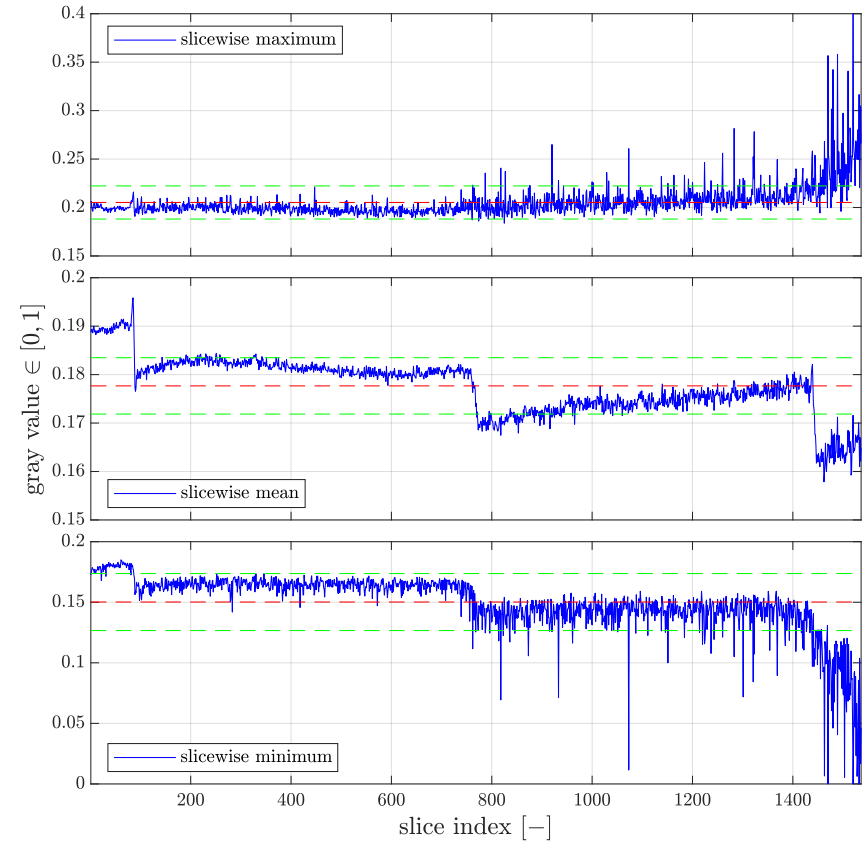


(f) for overlap divisor ol3

Figure A.1.: numerical studies of the gray value trends along the slice's stack height for $M \stackrel{(5.1)}{=} 4.95$ (continued)

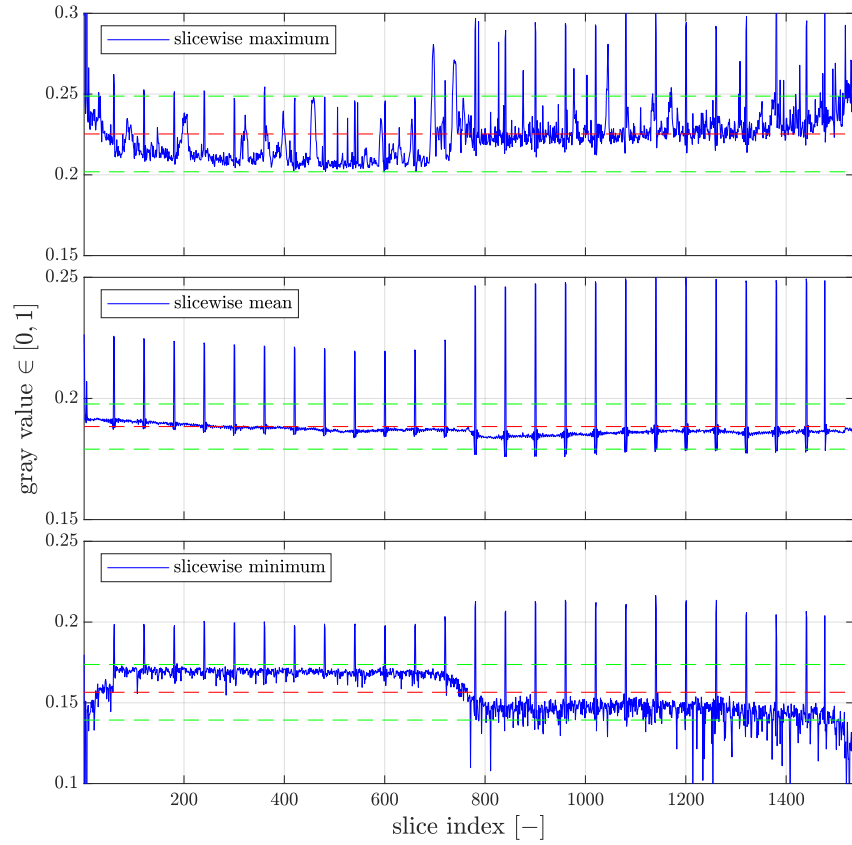
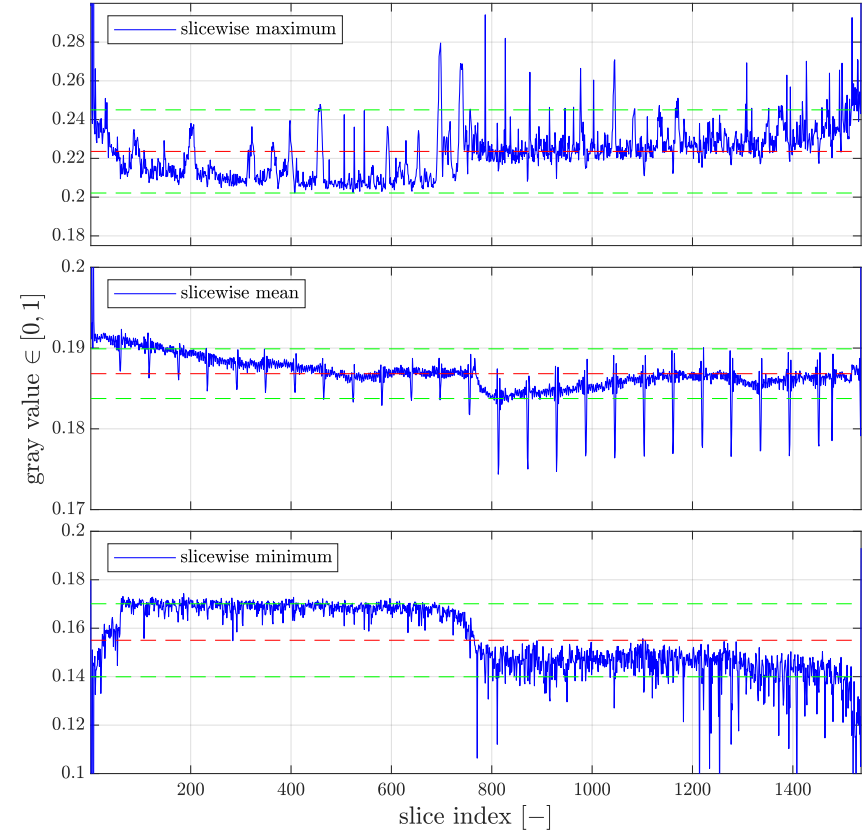


(g) for overlap divisor ol2



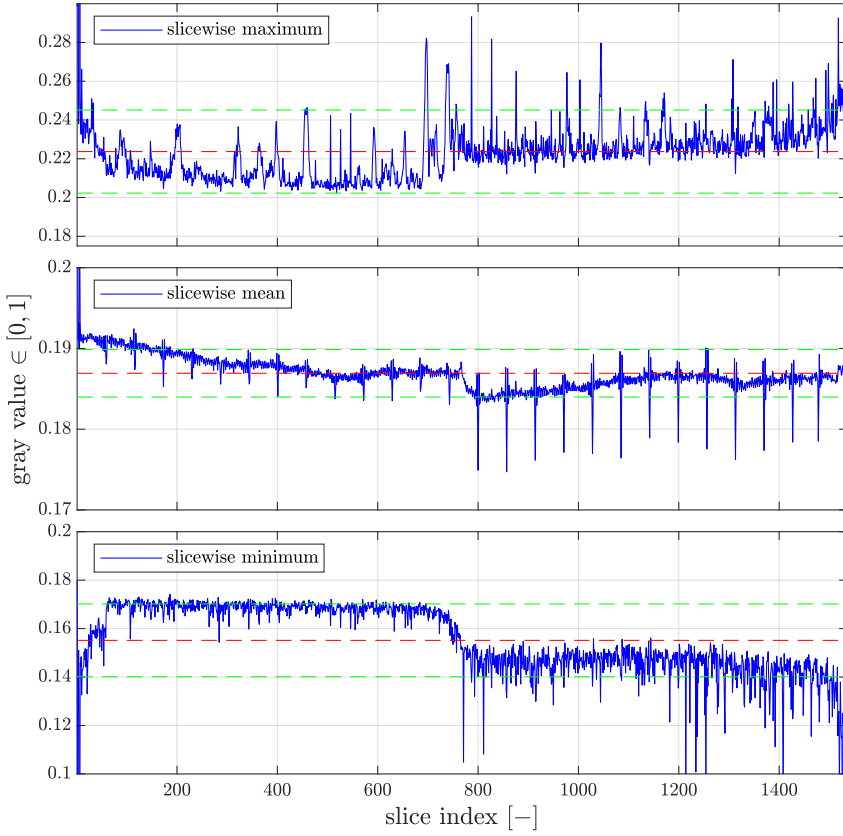
(h) 2D reconstruction (for comparison)

Figure A.1.: numerical studies of the gray value trends along the slice's stack height for $M^{(5,1)} \stackrel{=}{=} 4.95$ (continued)

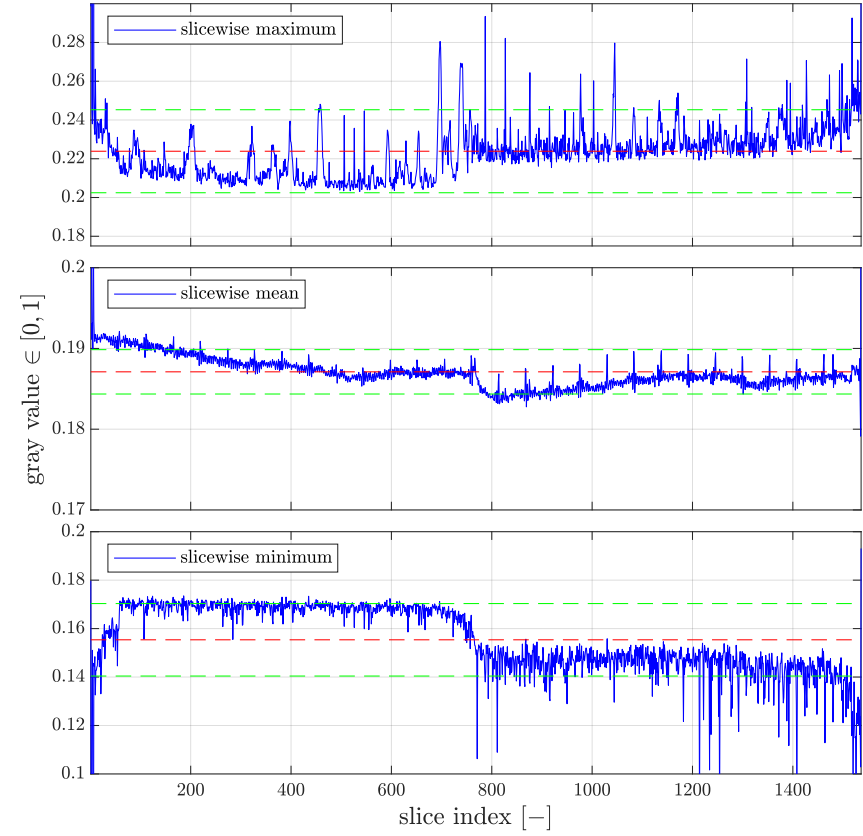
(a) without applying any overlap ("ol ∞ ")

(b) for overlap divisor ol50

Figure A.2.: numerical studies of the gray value trends along the slice's stack height for $M^{(5.1)} = 16.5$

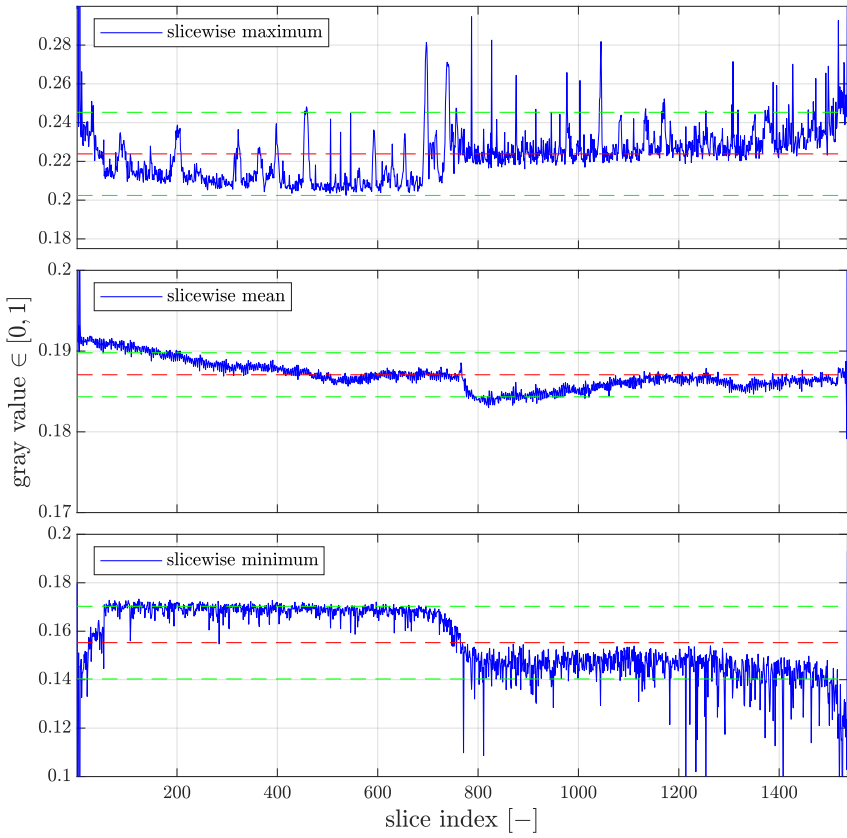


(c) for overlap divisor ol20

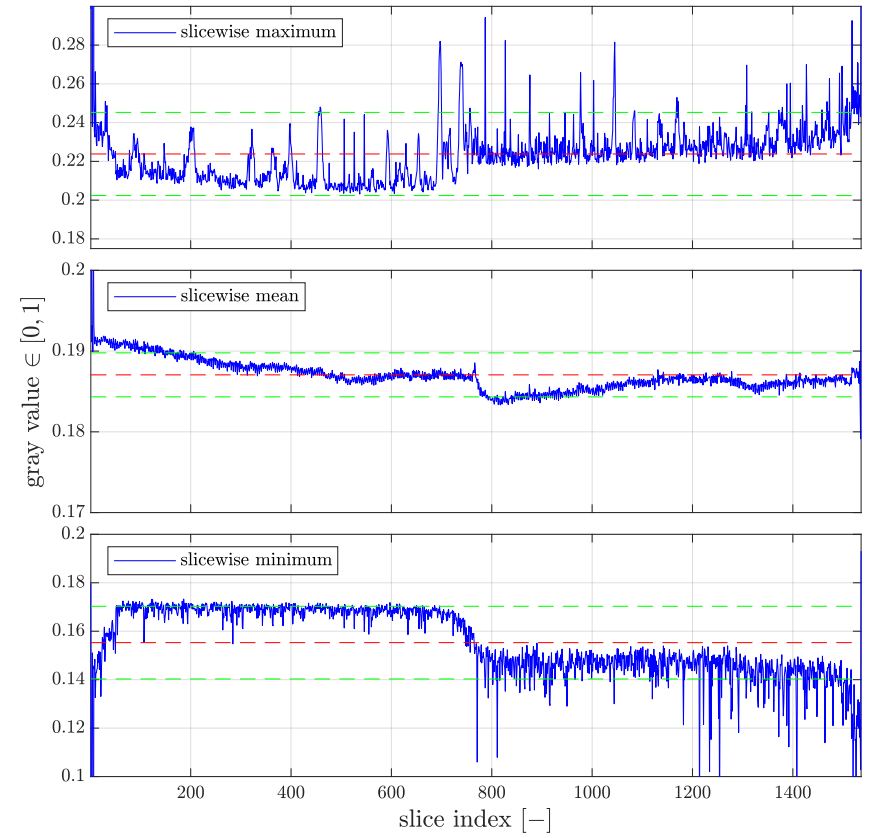


(d) for overlap divisor ol10

Figure A.2.: numerical studies of the gray value trends along the slice's stack height for $M^{(5,1)} = 16.5$ (continued)

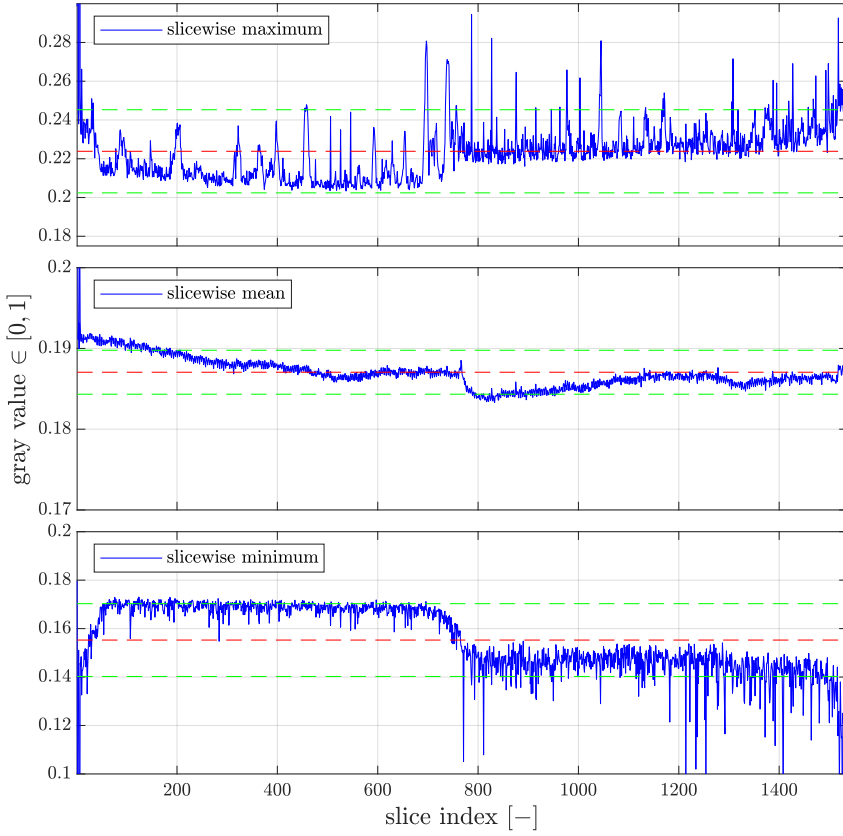


(e) for overlap divisor ol5

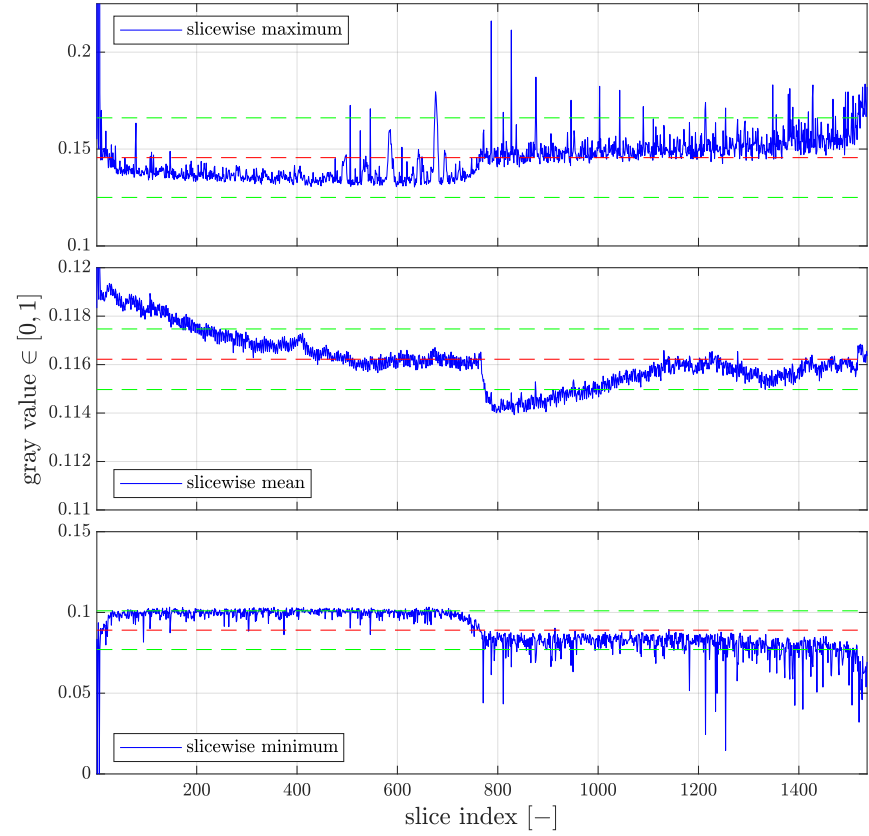


(f) for overlap divisor ol3

Figure A.2.: numerical studies of the gray value trends along the slice's stack height for $M^{(5.1)} \equiv 16.5$ (continued)

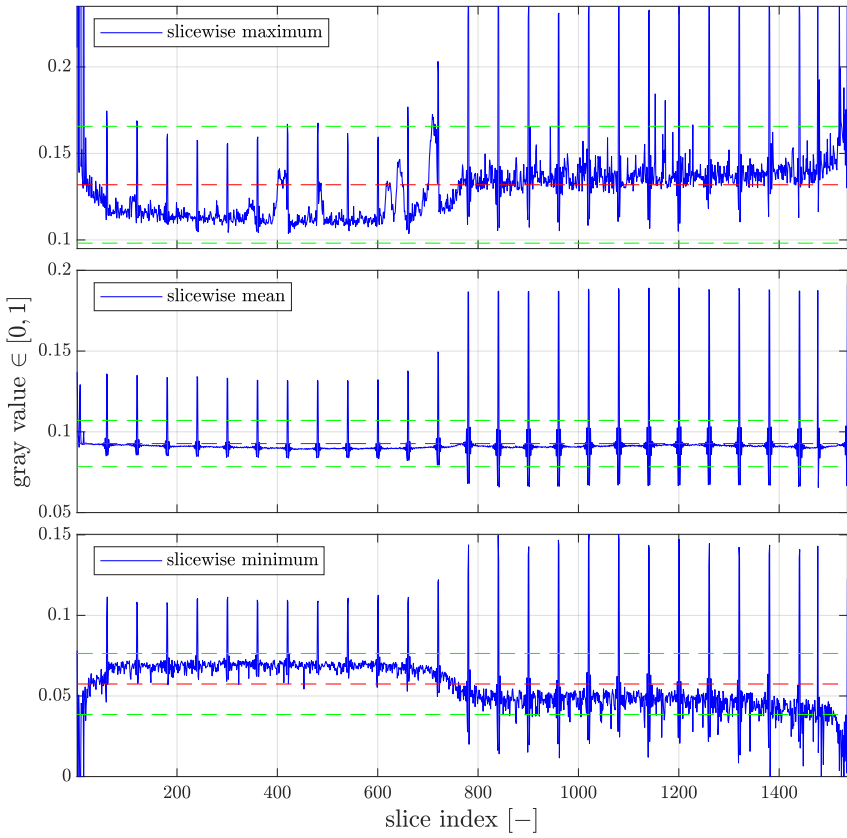


(g) for overlap divisor ol2

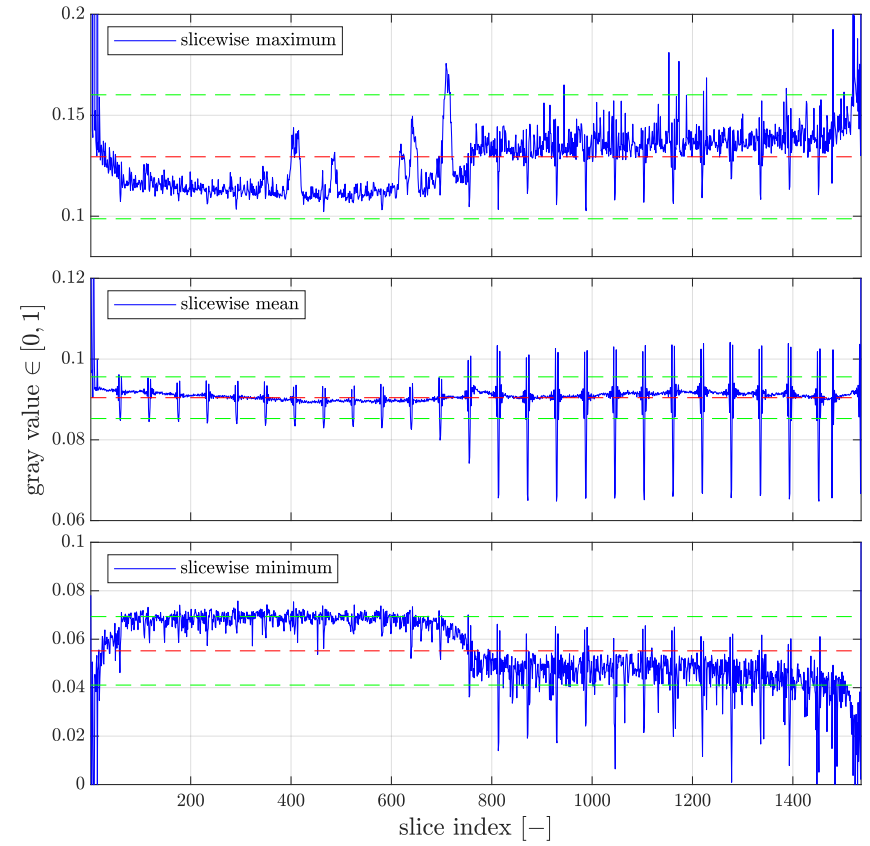


(h) 2D reconstruction (for comparison)

Figure A.2.: numerical studies of the gray value trends along the slice's stack height for $M^{(5,1)} = 16.5$ (continued)

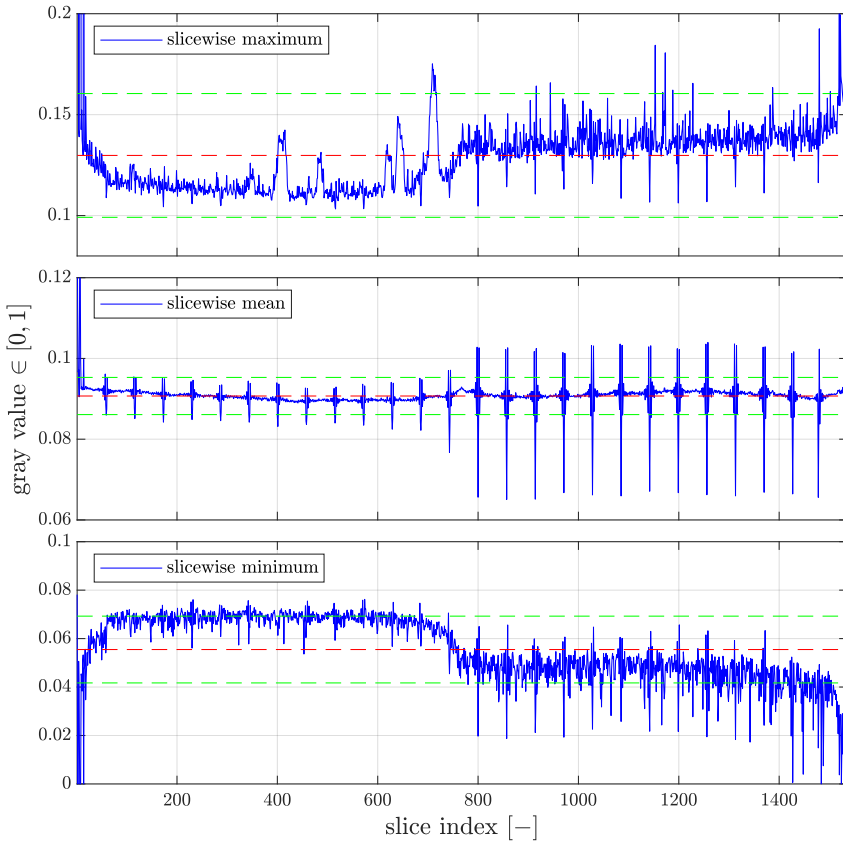


(a) without applying any overlap ("ol ∞ ")

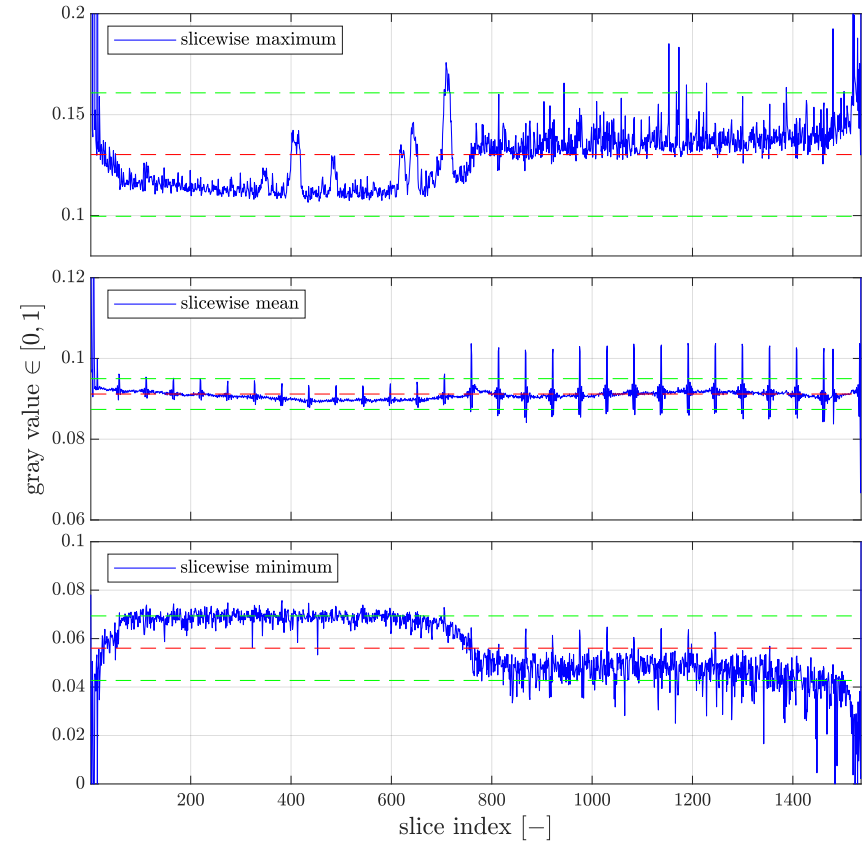


(b) for overlap divisor ol50

Figure A.3.: numerical studies of the gray value trends along the slice's stack height for $M \stackrel{(5.1)}{=} 33.02$

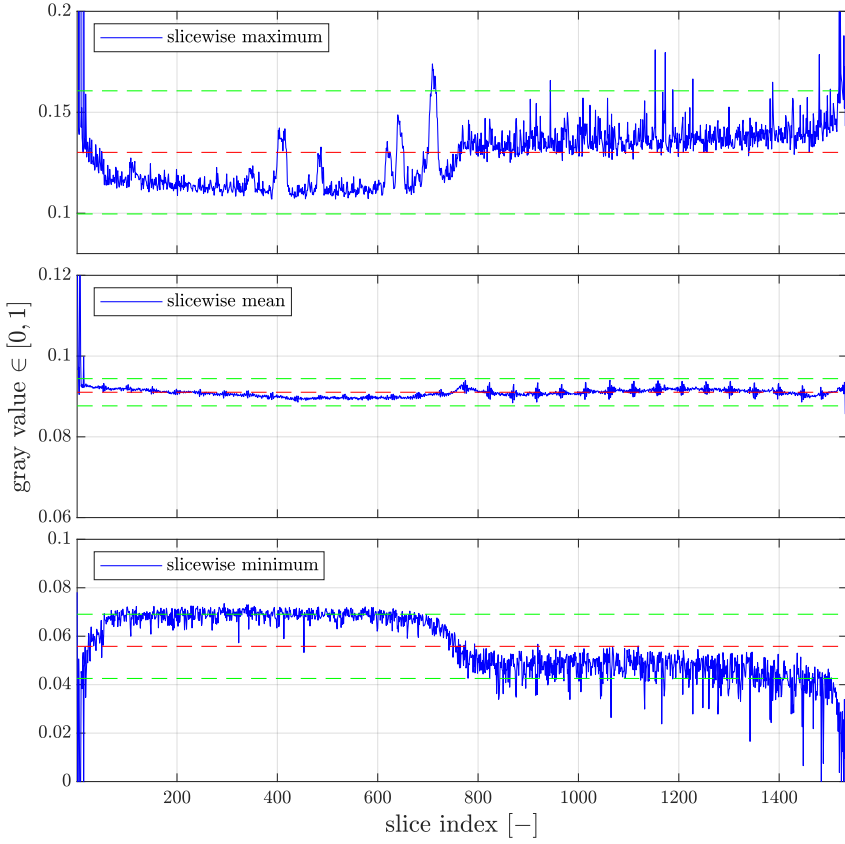


(c) for overlap divisor ol20

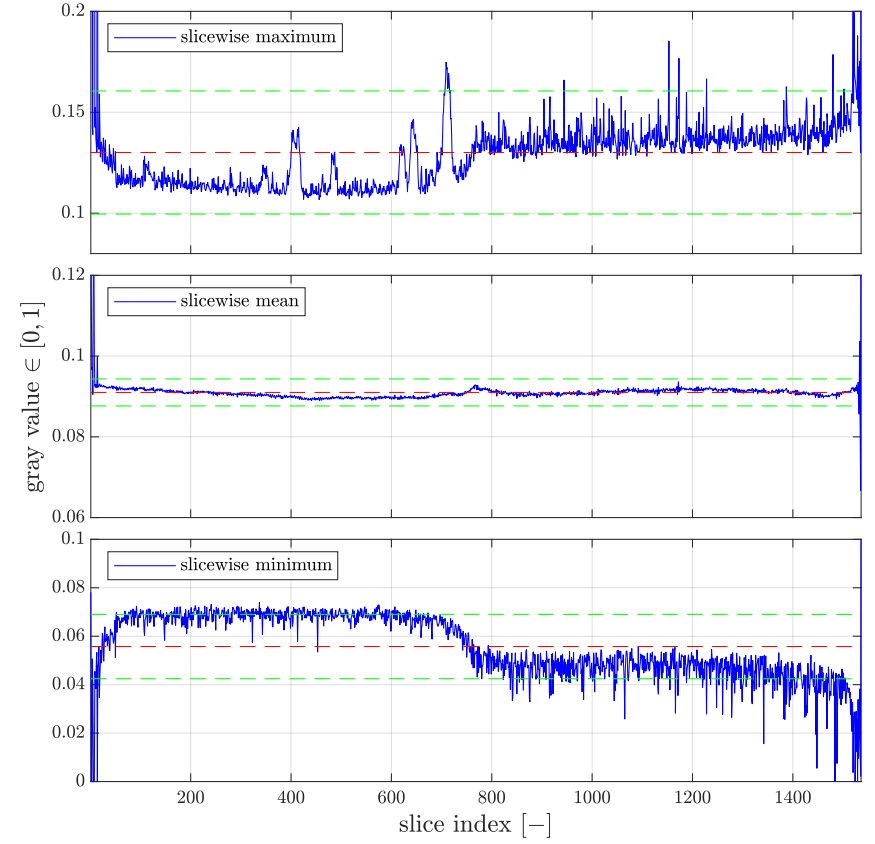


(d) for overlap divisor ol10

Figure A.3.: numerical studies of the gray value trends along the slice's stack height for $M \stackrel{(5.1)}{=} 33.02$ (continued)

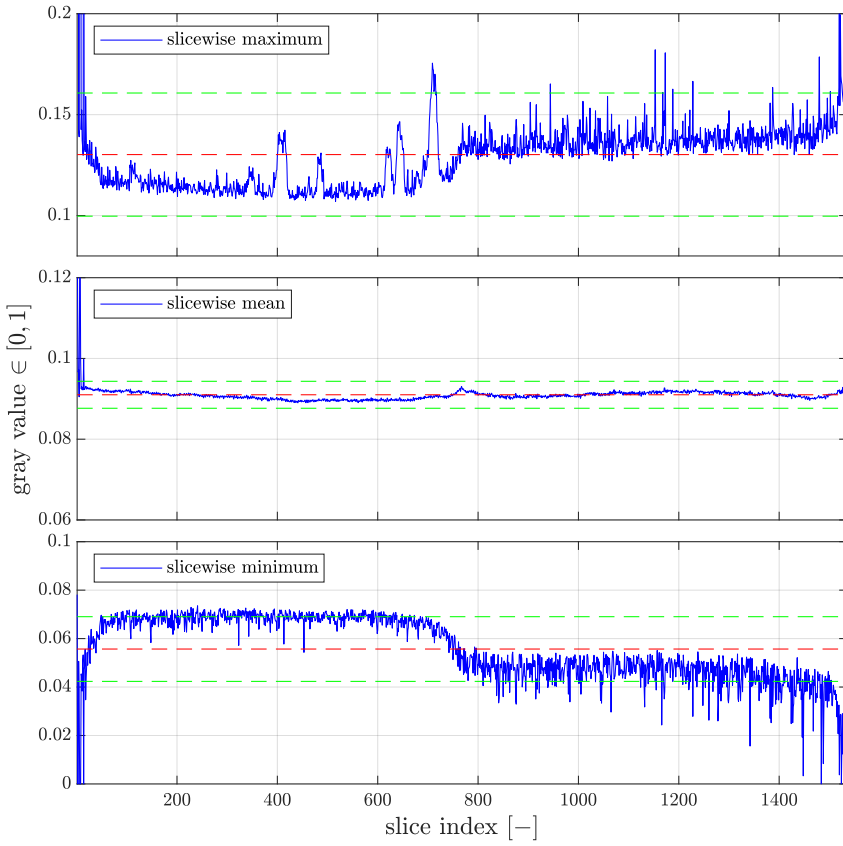


(e) for overlap divisor ol5

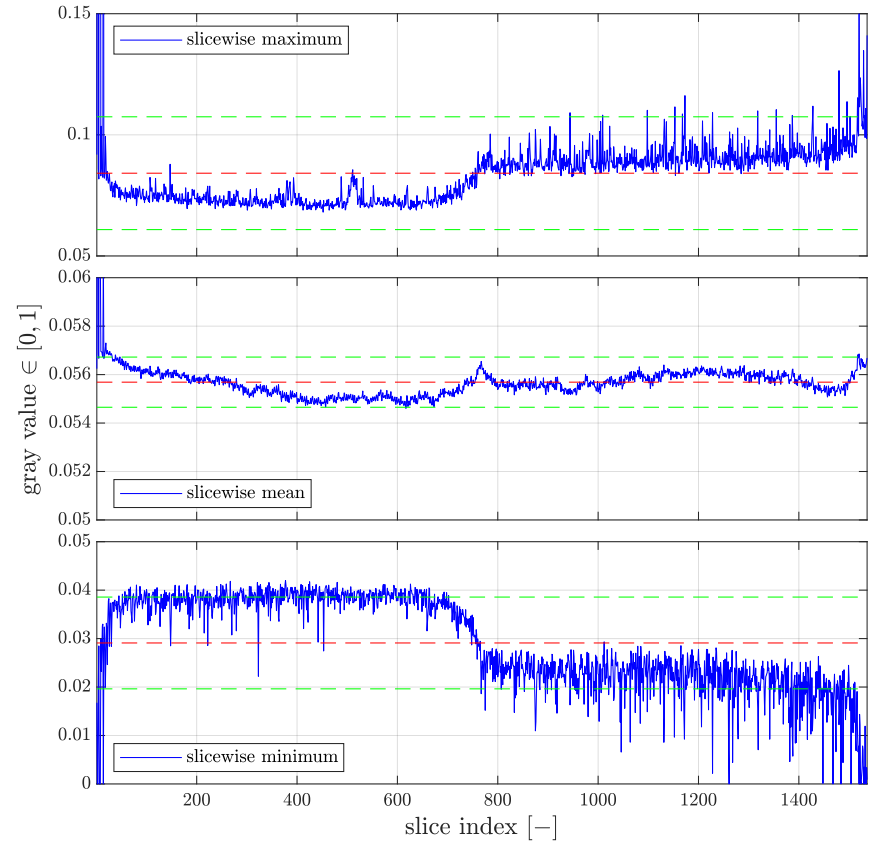


(f) for overlap divisor ol3

Figure A.3.: numerical studies of the gray value trends along the slice's stack height for $M^{(5.1)} \stackrel{=}{=} 33.02$ (continued)



(g) for overlap divisor ol2



(h) 2D reconstruction (for comparison)

Figure A.3.: numerical studies of the gray value trends along the slice's stack height for $M \stackrel{(5.1)}{=} 33.02$ (continued)

Bibliography

- Bale, H., Blacklock, M., Begley, M. R., Marshall, D. B., Cox, B. N., & Ritchie, R. O. (2011). Characterizing three-dimensional textile ceramic composites using synchrotron x-ray micro-computed tomography. *J. Am. Ceram. Soc.* 95, 392–402. doi:10.1111/j.1551-2916.2011.04802.x
- Biguri, A., Dosanjh, M., Hancock, S., & Soleimani, M. (2016). TIGRE: A MATLAB-GPU toolbox for CBCT image reconstruction. *Biomed. Phys. Eng. Express*, 2. doi:10.1088/2057-1976/2/5/055010
- Buzug, T. M. (2008). *Computed tomography*. Springer Berlin Heidelberg. doi:10.1007/978-3-540-74658-4_16
- Carmignato, S., Dewulf, W., & Leach, R. (2018). *Industrial x-ray computed tomography*. Springer International Publishing. doi:10.1007/978-3-319-59573-3
- Cnudde, V. & Boone, M. N. (2013). High-resolution x-ray computed tomography in geosciences: A review of the current technology and applications. *Earth Sci. Rev.* 123, 1–17. doi:10.1016/j.earscirev.2013.04.003
- DIN EN ISO 15708-3:2019-09, Zerstörungsfreie Prüfung – Durchstrahlungsverfahren für Computertomographie – Teil 3: Durchführung und Auswertung (ISO 15708-3:2017); Deutsche Fassung EN ISO 15708-3:2019. (2019). Beuth Verlag GmbH. doi:10.31030/3054744
- Feldkamp, L. A., Davis, L. C., & Kress, J. W. (1984). Practical cone-beam algorithm. *J. Opt. Soc. Am. A*, 1, 612–619. doi:10.1364/JOSAA.1.000612
- Ferrucci, M., Leach, R. K., Giusca, C., Carmignato, S., & Dewulf, W. (2015). Towards geometrical calibration of x-ray computed tomography systems – a review. *Meas. Sci. Technol.* 26. doi:10.1088/0957-0233/26/9/092003
- Gregor, J. & Benson, T. (2008). Computational analysis and improvement of SIRT. *IEEE Trans. Med. Imag.* 27, 918–924. doi:10.1109/TMI.2008.923696
- Gürsoy, D., Carlo, F. D., Xiao, X., & Jacobsen, C. (2014). TomoPy: A framework for the analysis of synchrotron tomographic data. *J. Synchrotron Radiat.* 21, 1188–1193. doi:10.1107/S1600577514013939
- Kak, A. C. & Slaney, M. (2001). *Principles of computerized tomographic imaging*. Society for Industrial and Applied Mathematics. doi:10.1118/1.1455742
- Kasban, H., El-Bendary, M. A. M., & Salama, D. H. (2015). A comparative study of medical imaging techniques. *International Journal of Information Science and Intelligent Systems*, 4, 37–58.
- Klingen, B. (2001). *Fouriertransformation für ingenieur- und naturwissenschaften*. Springer Berlin Heidelberg. doi:10.1007/978-3-642-56775-9
- Kloek, T. (2012). *Conjugate gradients and conjugate residuals type methods for solving least squares problems from tomography* (Bachelor's thesis, Technische Universiteit Delft).
- Louk, A. C. & Suparta, G. B. (2015). 3D image reconstruction on x-ray micro-computed tomography. In C. Quan, K. Qian, A. Asundi, & F. S. Chau (Eds.), *International conference on experimental mechanics 2014* (Vol. 9302, pp. 131–137). SPIE. doi:10.1117/12.2081193
- Mirone, A., Gouillart, E., Brun, E., Tafforeau, P., & Kieffer, J. (2014). The PyHST2 hybrid distributed code for high speed tomographic reconstruction with iterative reconstruction and a priori knowledge capabilities. *Nucl. Instrum. Methods Phys. Res., Sect. B*, 324, 41–48. doi:10.1016/j.nimb.2013.09.030
- Octopus reconstruction user manual*. (2018). Version 8.9.4. Inside Matters.

- Palenstijn, W. J., Batenburg, K. J., & Sijbers, J. (2011). Performance improvements for iterative electron tomography reconstruction using graphics processing units (GPUs). *J. Struct. Biol.* 176, 250–253. doi:10.1016/j.jsb.2011.07.017
- Poludniowski, G., Landry, G., DeBlois, F., Evans, P. M., & Verhaegen, F. (2009). Spekcalc: A program to calculate photon spectra from tungsten anode x-ray tubes. *Phys. Med. Biol.* 54, N433–N438. doi:10.1088/0031-9155/54/19/N01
- Ramachandran, G. N. & Lakshminarayanan, A. V. (1971). Three-dimensional reconstruction from radiographs and electron micrographs: Application of convolutions instead of fourier transforms. *Proc. Natl. Acad. Sci. U.S.A.* 68, 2236–2240. doi:10.1073/pnas.68.9.2236
- Ruf, M. & Steeb, H. (2020a). An open, modular and flexible micro-XRCT system for research. *To be submitted to Rev. Sci. Instrum.*
- Ruf, M. & Steeb, H. (2020b). Micro-XRCT data set of open-pored asphalt concrete. DaRUS. doi:10.18419/darus-639
- Shepp, L. A. & Logan, B. F. (1974). The fourier reconstruction of a head section. *IEEE Trans. Nucl. Sci.* 21, 21–43. doi:10.1109/TNS.1974.6499235
- Stock, S. R. (2008). *Microcomputed tomography*. CRC Press. doi:10.1201/9780429186745
- Sun, W., Brown, S. B., & Leach, R. K. (2012). An overview of industrial x-ray computed tomography.
- van Aarle, W., Palenstijn, W. J., Beenhouwer, J. D., Altantzis, T., Bals, S., Batenburg, K. J., & Sijbers, J. (2015). The ASTRA toolbox: A platform for advanced algorithm development in electron tomography. *Ultramicroscopy*, 157, 35–47. doi:10.1016/j.ultramic.2015.05.002
- van Aarle, W., Palenstijn, W. J., Cant, J., Janssens, E., Bleichrodt, F., Dabavolski, A., . . . Sijbers, J. (2016). Fast and flexible x-ray tomography using the ASTRA toolbox. *Opt. Express*, 24. doi:10.1364/OE.24.025129
- Vlassenbroeck, J., Dierick, M., Masschaele, B., Cnudde, V., Hoorebeke, L. V., & Jacobs, P. (2007). Software tools for quantification of x-ray microtomography at the UGCT. *Nucl. Instrum. Methods Phys. Res., Sect. A*, 580, 442–445. doi:10.1016/j.nima.2007.05.073