



Extension and Evaluation of Tissue Classification Based on a Backscattering Model for 3D Ultrasound Computer Tomography

Master Thesis

submitted by

Benedikt Ebener

Institute for Data Processing and Electronics
Karlsruhe Institute of Technology

Reviewer: PD Dr. habil. Nicole Ruiter
Second reviewer: Prof. Dr. Marc Weber
Advisor: Dr. Torsten Hopp

5th March 2021

Statutory declaration

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

Karlsruhe, 5th March 2021

B.Ebener

Abstract

The three dimensional ultrasound computer tomography (USCT) is a novel imaging technique for the detection of early stage breast cancer in women. The combination of the basic principle of ultrasonic imaging with the computed tomography (CT) allows the reconstruction of high resolution three dimensional images of the examined breast tissue. Currently there are three modalities in use: reflection imaging, attenuation imaging and speed of sound transmission imaging. For the reflection image a 3D synthetic aperture focusing technique (SAFT) is applied to calculate a reflectivity value for each voxel of the image.

In this thesis new imaging techniques are proposed which allow for the preservation of directional information of reflections. With the analysis of the reflection characteristics of the tissue a distinction of tissue types with a specular reflectivity or diffuse reflecting tissue types is aimed for.

In a previous work a prototypical implementation of a method for the analysis of the directional information was conducted, in which only a four dimensional image had been captured. This thesis extends the extraction of directional information into the five dimensional space by considering incoming and outgoing directions separately.

A generalisation of the discretisation problem of incoming and outgoing directions is introduced which allows to find a trade-off between the memory consumption and resolution of the angular segmentation. Furthermore, a more efficient method for the assignment of vectors to the discretised directional segmentation is presented which can decrease the execution time by multiple orders. The developed methods have been fully integrated into the clinically used image reconstruction software, which also allows to perform a sound speed correction to increase the resolution and contrast of the image.

The methods have been evaluated with experimental data. The process of assigning the reflectivity information to individual incoming and outgoing directions has been successfully verified. An analysis of the reflection characteristics for distinct material types show that these materials can be distinguished. For the three dimensional visualisation of the five dimensional results different methods are presented. Furthermore, it is shown that the sound speed correction makes the differentiation of material types more distinct.

This thesis lays the foundation for in-depth analysis of reflection characteristics of different breast tissue and may provide the tools for the detailed classification of tissue types for the early detection of breast cancer.

Acknowledgments

Ich möchte die Gelegenheit nutzen und den vielen Menschen danken, die mich bei meiner Masterarbeit und meinem Studium unterstützt haben. Mein besonderer Dank gilt meinem Betreuer Torsten Hopp für die viele Zeit und die Mühen, die er aufgebracht hat, um mich bei den ganzen kleinen und großen Problemen während der Masterarbeit zu unterstützen. Darüber hinaus möchte ich mich für sein großartiges Feedback während der Korrekturphase bedanken. Bei Michael Zapf möchte ich mich ebenfalls bedanken für die Zeit, die er sich genommen hat und die vielen tollen Ideen, die zu großartigen Ergebnissen geführt haben. Ebenfalls möchte ich mich bei Nicole Ruiter für die herzliche Unterstützung und ihren Rat bedanken. Allen Mitarbeitern und Doktoranden am IPE möchte ich für die schöne gemeinsame Zeit danken, die wir trotz der Pandemiesituation miteinander verbringen durften. Nicht zuletzt möchte ich mich bei meinen Eltern Barbara und Frank für die emotionale, finanzielle und kulinarische Unterstützung während meines gesamten Studiums bedanken.

Contents

Abstract	i
Acknowledgments	iii
Abbreviations	vii
1 Introduction	1
1.1 Motivation and objectives	3
2 Fundamentals	5
2.1 3D ultrasound computer tomography (3D USCT)	5
2.1.1 Multi-modality	6
2.1.2 Synthetic aperture focusing technique (SAFT)	9
2.1.3 Speed of sound correction	13
2.2 Characterisation of reflections	14
2.3 Graphic processing unit (GPU)	16
2.3.1 CUDA-programming	17
3 Methods	21
3.1 Extension of the SAFT information dimensionality	21
3.2 Discretisation of directional information	25
3.2.1 Platonic solids	25
3.2.2 Arbitrary directional segmentation	27
3.3 Comparison vectors	30
3.4 General procedure for calculating directional information	31
3.5 Assignment to the directional dimensions	33
3.5.1 Orthogonality	33
3.5.2 Angle sorting method	36
3.5.3 Orthogonality threshold method	39
3.6 Performance implications of assignment approaches	44
4 Results	47
4.1 Experimental Setup	47
4.2 Evaluation of the assignment process during 5D reconstruction	50

4.3	Comparison of orthogonality threshold method and angle sorting method	54
4.4	Performance of the directional dimension assignment methods	59
4.5	Evaluation of the differentiation of tissue by directional information	63
4.6	Angular relation between directional information	66
4.7	Deviation imaging & Maximum imaging	70
4.8	Influence of the speed of sound correction	74
5	Discussion & Outlook	79
5.1	Assessment of four and five dimensional approach	79
5.2	Resolution of the discretisation of direction	82
5.3	Comparison of the assignment algorithms	82
5.4	Interpretation of visualisation of the reflection characteristics	84
5.5	Discussion of the implementation of the methods	84
5.6	Conclusion & Outlook	85
A	Appendix	87
A.1	Alternative deduction of orthogonality approach	87
A.2	Spherical coordinate system	89
References		91

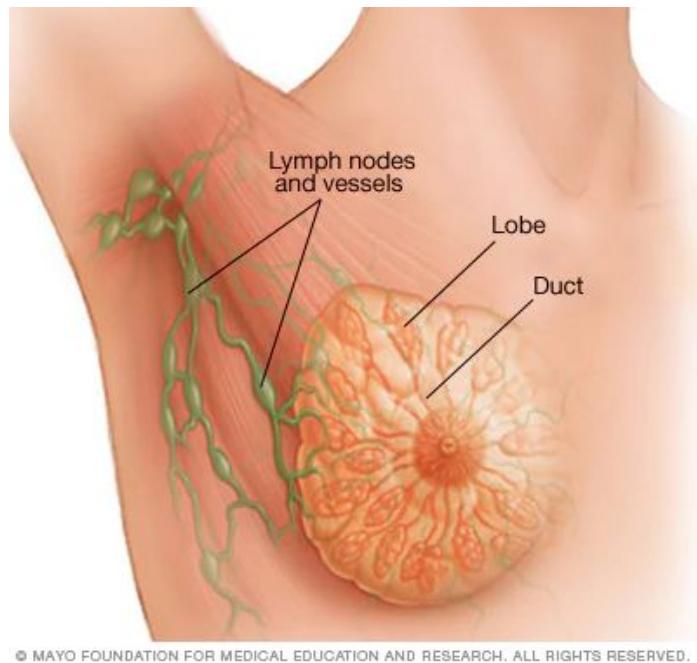
Abbreviations

ALU	arithmetic logic unit
A-scan	amplitude scan
CT	computed tomography
GPU	graphic processing unit
SAFT	synthetic aperture focusing technique
SIMD	single instruction, multiple data
SM	streaming module
SOS	speed of sound
TAS	transducer array system
TOF	time of flight
USCT	ultrasound computer tomography

Introduction

The last couple of decades have shown a steady increase of the incidence for female breast cancer. In 1935 the incidence for breast cancer was at 16.3 diagnoses per 100.000 women whereas 38.5 per 100.000 women were diagnosed in the year 2015 [1]. In Germany 71.600 women were diagnosed with breast cancer in the year 2013 with cases having doubled since 1970. A more fortunate development has been observed in the death rate in patients diagnosed with breast cancer. Since 1999 the mortality rate decreased by about a third in patients under 50 and was 25% lower in patients between the age 50-69 during that period. One of the reasons that the mortality rate could be decreased is the introduction of the screening program for women between the ages 50-69 [2]. Since then the number of diagnoses of advanced tumour stages were decreased [3]. The prognosis for the treatment of cancer depends on the stage of the cancer and how early it was discovered, as with the size of the tumour the risk of spreading into other organs increases. As the mortality rate directly correlates with the tumour size and stage of the cancer, an early detection of cancerogenous tissue is one of the most effective ways to increase the survival rate of patients [4], [5].

The female breast consists of mainly fatty tissue and the mammary glands also known as lobules which can produce milk. Additionally, the ducts (see Figure 1.1) are part of the breast tissue which form the transport channels for the milk between the lobules and the nipple.



© MAYO FOUNDATION FOR MEDICAL EDUCATION AND RESEARCH. ALL RIGHTS RESERVED.

Figure 1.1: Schematic of the female breast. The lobes form the milk producing system of the breast whereas the ducts are channels for the milk towards the nipple of the breast. Source: [6].

The new imaging technique ultrasound computer tomography (USCT) uses the basic principles of the ultrasound imaging technique which are extended by some principle elements of the the computed tomography (CT). For the measurement an ultrasound transducer emits the ultrasound waves into the imaging aperture of the USCT-system. The receivers record the pressure over time and yield a so-called amplitude scan (A-scan). With the combination of the A-scans yielded from all emitter-receiver-combinations, a reflection and transmission images of the examined tissue can be reconstructed. The combination of multiple modalities like the reflection and speed of sound for the imaging can be used to generate high resolution images.

Compared to the widely used screening method, the X-ray mammography, which uses ionizing radiation, the three dimensional USCT has advantages and makes this novel imaging technique a promising alternative: By utilizing the ultrasound, high resolution 3D images of the breast tissue can be yielded without the use of ionizing radiation. Furthermore, the examination process with an USCT is much more comfortable for the patient since the breast does not have to be deformed during the imaging procedure.

1.1 Motivation and objectives

Until now reflectivity imaging calculates the qualitative average of the reflectivity of the tissue, however the scattering characteristics of the tissue can not be characterised from these data. It is assumed that tumour tissue consists of an inhomogeneous distribution of cancer cells which results in a distinct backscattering behaviour of this particular tissue. Similar to optical approaches it can further be assumed that small particles in the tissue have a omnidirectional reflection characteristic like a point scatterer. Tissue types with a predominant amount of inhomogeneously distributed small particles are therefore assumed to have diffuse reflection properties whereas larger, even tissue patches with homogeneous structure have a more specular reflection characteristic. By introducing the methods for evaluating the reflection characteristic in the reconstruction algorithm, the reflection properties of different tissue types can be analysed.

Previous approaches resulted in a four dimensional image. The limitations of the four dimensional approach is that the actual back scattering characteristic can not be assessed without a 2nd directional information. The 4th dimension holds the information of the voxel-receiver configuration. With the 5th dimension also the voxel-emitter relation can be regarded and used to analyse the reflection characteristics of the tissue. These limitations were tackled by extending the algorithm of the previous implementation into a fifth dimension. Thus, not only the scalar information about the outgoing ultrasound wave from the voxel to the receiver is regarded but also the direction of the incoming pulse from the emitter to the voxel. The directional information between the fourth and fifth dimension will also allow for the analysis of reflection properties of the tissue concerning its specular and diffuse reflection parts. This allows the determination of a specific reflection characteristic of the tissue.

The usage of platonic solids for the generation of directional vectors to discretise the directional information of the voxel in the work of Patrick Hucker [7] limited the resolution of the directional vectors and with that the capabilities to distinguish different reflection types. The generalisation of the discretisation problem of the measurement volume from small platonic solids with a limited number of directional vectors to the generation of arbitrary number of directional vectors with a higher resolution of the discretisation will increases the capabilities of the reflection analysis further. A simple example for different reflection types is given in Figure 1.2:

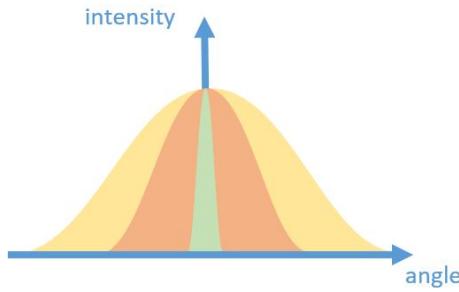


Figure 1.2: Example of three different types of reflection characteristics. The green bell curve shows close to specular reflectivity whereas the orange and yellow curves have a more diffuse reflection characteristic.

To distinguish between the different reflection types it is necessary to reach a high enough discretisation of the direction information. In this thesis a new approach for the arbitrary extension of the resolution of the discretisation of the directional information is proposed, which overcomes the aforementioned limitations.

Previous implementations for the analysis of the reflection characteristic lacked the capabilities of regarding the different speed of sound (SOS) of tissues during the reconstruction. At that time a constant sound speed was assumed. Since then the calculation of the SOS for the different areas of the measurement aperture was introduced and the quality of the final image could be increased. In this work a SOS-correction will be introduced to the reconstruction algorithm and its influence on the resulting reflection characteristics compared to non-SOS-corrected results will be assessed. Furthermore, the only prototypical implementation of the previous work as well as the new insights of this thesis were implemented in the clinically used reconstruction algorithm of the USCT-project with regard to downward-compatibility and data structure.

Fundamentals

The technical fundamentals for the creation of high resolution three dimensional images are presented on the following chapters. Since the main objective of this thesis is the classification of different tissue types in reference to their reflection characteristics the basic reflection types are emphasised in this chapter.

2.1 3D ultrasound computer tomography (3D USCT)

The three dimensional ultrasound computer tomography (3D USCT) is a imaging technique which uses the basic principle of the sonography. The semi-ellipsoidal aperture allows for more imaging techniques than the standard ultrasound sonography does. During the measurement unfocused and approximately spherical ultrasound waves are emitted into the aperture of the USCT. The waves interact with the tissue, the coupling medium and the scanner itself. These interactions lead to transmission paths, reflection paths and certain attenuation along the trajectory of the wave. The interaction of the sound wave with the tissue leads to a distinct pattern of reflection and attenuation that allows to generate several images from the recorded A-scans. With these imaging capabilities the USCT imaging technique became a promising new technique for the detection of breast cancer in early stages [8]. The second version prototype of the 3D USCT device can be seen in Figure 2.1.

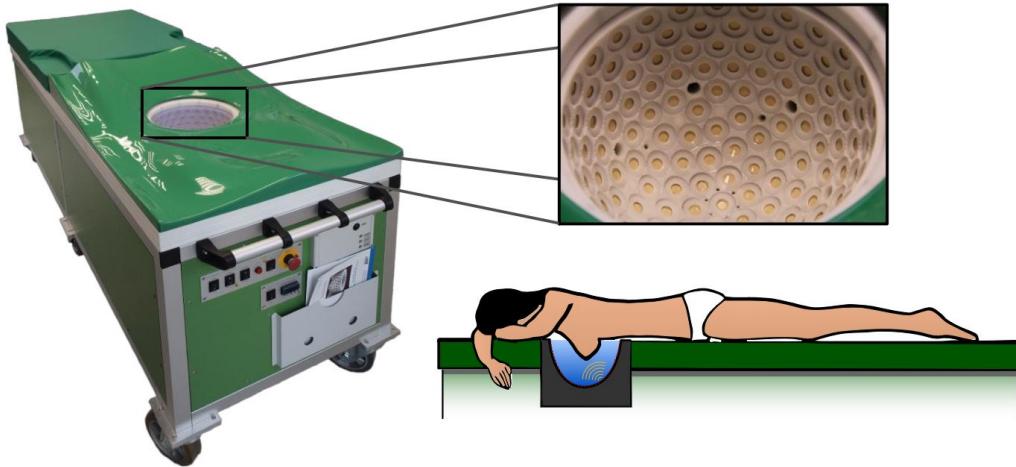


Figure 2.1: The prototype of an 3D-USCT device at the KIT research facility. Source: [9].

The picture on the left shows the patient bed with the imaging aperture. The patients have to lie down on their stomach during the imaging procedure and have to remain as still as possible to avoid image degradation [8]. One breast at a time will be placed in the water-filled semi-ellipsoidal imaging aperture. On the top-right of Figure 2.1 the aperture is shown with the ultrasound transducer elements mounted in the wall of the aperture.

To measure the pressure over time, the prototype of the USCT device at the KIT comprises 157 transducer array systems (TASs). Each TAS consists of four emitting transducers and nine receiving transducers. Overall, 628 emitting and 1413 receiving transducers and ten aperture rotation positions are used to record A-scans to reconstruct an three dimensional images. Currently, ten positions are used, which was regarded as a good trade-off between the sparseness and the acquisition time. The aperture itself has a height of 17 cm and diameter of 26 cm [9].

2.1.1 Multi-modality

Three modalities for the reconstruction of a high resolution 3D-USCT image are currently in use: reflectivity, speed of sound and attenuation [10].

For the **Attenuation imaging** the A-scans of two transducers are compared. The sound wave interacts with the tissue and the coupling medium and loses intensity depending on the property of the tissue on that particular trajectory. The density of respective tissue influences the scattering and absorption of the sound waves energy. To quantify the attenuation a reference measurement of the empty imaging aperture precedes the actual acquisition of medical data. The amplitudes of the empty measurement are then compared to the levels with the breast in place. With an algebraic reconstruction technique the final attenuation-image

can be reconstructed [11]. An example is given in Figure 2.2. The attenuation is given with the unit of attenuation in dB over the frequency of the ultrasound and the distance. The image shows the attenuation map of a breast phantom in the USCT. The bright spot in the middle is an artificial tumour. The brighter the colour of the tissue is depicted the higher is the attenuation of that part of the tissue.

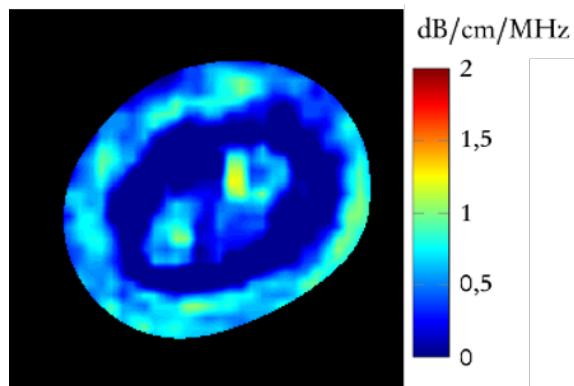


Figure 2.2: Example of an attenuation image reconstructed by an algebraic reconstruction technique.

The **SOS imaging** technique is used to quantify the tissue specific sound speed. For that, the A-scans of emitter-receiver-combinations are used for which the transducers are placed in the opposite site of the imaging aperture so that the sound waves have to pass through the tissue and the detected signals do not primarily arise from reflections. The comparison of the transmission pulse of the A-scan to the time of emission leads to a value for the propagation time and hence the SOS along the path from one transducer to the other. To get a 3D image the measurement has to be repeated for a multitude of emitter-receiver-combinations around the aperture. In [12] it was shown that with the USCT in transmission mode cysts in the female breast can be distinguished from surrounding tissue by the difference in sound speed in the different types of tissue. It was shown that normal tissue has a SOS of about $1400 - 1450 \text{ m/s}$ and tumorous tissue a SOS of $1500 - 1520 \text{ m/s}$. Factors that influence the speed of sound in different types of tissue are the density, the micro-structure, temperature and elasticity of the tissue. Compared to the surrounding tissue the physical properties of tumours differ from the normal case and lead often to a higher speed of sound. Reconstructing the transmission image with an algebraic reconstruction technique helps classifying different tissue types by the means of their sound speed. An example for the same breast phantom as above is shown in Figure 2.3. In this case the spot in the middle of the image shows a high speed of sound where the artificial tumour is located.

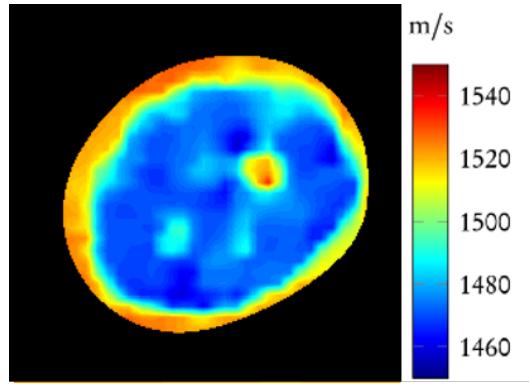


Figure 2.3: Example of a speed of sound map reconstructed by an algebraic reconstruction technique.

For the **Reflectivity imaging** technique one emitting transducer sends an ultrasound wave into the aperture and all receivers within an angle of approximately $0^\circ - 120^\circ$ between the sender normal and the receiver normal record the pressure over time. The measurement setup for one set of A-scans (i.e. for one emitter) can be seen in Figure 2.4. Hence, not all receivers are used in this set of A-scans. The image shows the aperture from above with a tilted angle. All the active receivers are depicted in green for one emitter. On the opposite side of the group of emitters there is a gap where the A-scans of the receivers are not regarded. For the reflection imaging the values of these receivers have proven to lower the signal to noise ratio of the image and therefore are not regarded.

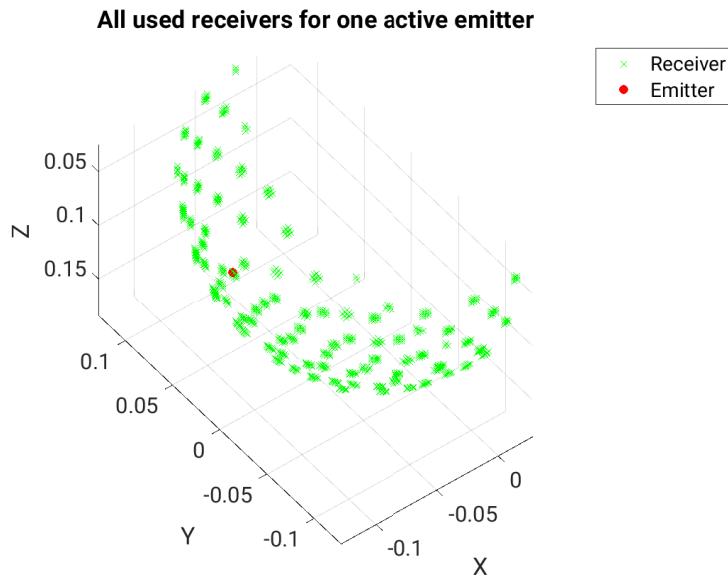


Figure 2.4: The configuration of receivers for one active emitter. All used receivers are shown in green, the active emitter is plotted in red. The units are given in meters.

The breast phantom example for the reflection image is given in Figure 2.5. The resolution of the reflection image is much higher compared to the SOS map and the attenuation map. The information of the other two images can be combined to improve the resolution of the reflection image even further by providing information about local attenuation properties of the tissue as well as the sound speed can be regarded during the reconstruction of the reflection image.

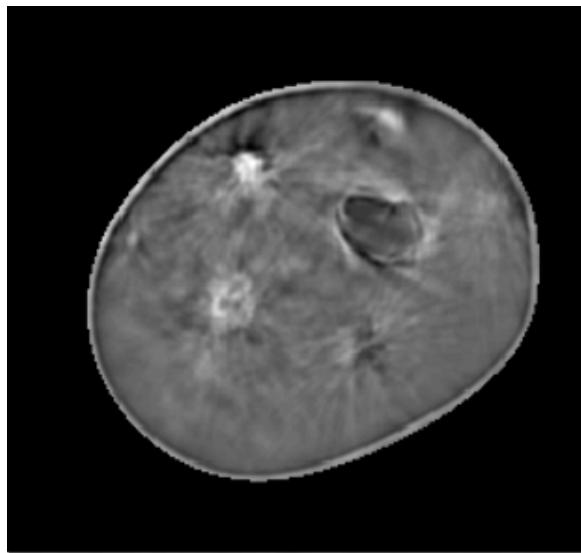


Figure 2.5: Example of a reconstructed reflection image of the breast phantom.

For the reconstruction of the reflection image the synthetic aperture focusing technique (SAFT) is used. Since this thesis is mainly focused on the reconstruction of USCT reflection imaging the following section 2.1.2 describes how SAFT can be used to extract all the necessary information from the A-scans to reconstruct a high resolution reflection image.

2.1.2 Synthetic aperture focusing technique (SAFT)

The SAFT is used to reconstruct reflection images from the raw data of ultrasound imaging. By combining the data of the A-scans from a multitude of emitter-receiver combinations it is possible to reconstruct a high resolution image with sub-millimetre-resolution.

Figure 2.6 shows the principle of an A-scan. The 3D volume of the aperture is shown on the left side. For this example only two of the many transducers that normally are placed on the wall of the aperture are shown. On the right the plot of the recorded A-scan at the receiver is shown by the graph. The actual A-scan consists of a discrete plot of the where the pressure $p(t)$ is plotted over the time t . The first pulse in the graph is the transmission pulse that reaches the receiving transducer. The second pulse originates from scattered waves at a certain location in the tissue and thus has a lower amplitude than the first pulse. The time

duration it takes for the pulse to propagate from the emitter to the point of scattering and then to the receiving transducer is called time of flight (TOF). One reason for the decreased amplitude of the reflected pulse compared to the transmission pulse is the longer path that the wave has to propagate. Scattering effects and reflections lead to the attenuation of the signal and therefore to a lower amplitude.

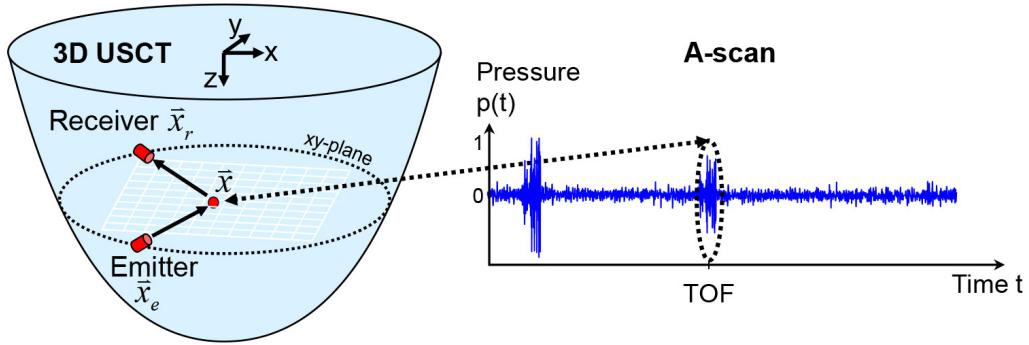


Figure 2.6: Simple example of an A-scan for one emitter-receiver combination. One transducer $\vec{\chi}_e$ emits a pulsed wavefront into the aperture. The A-scan shows the pressure over the time measured by the receiving transducer $\vec{\chi}_r$. The first pulse in the diagram is the pulse caused by the direct transmission of the emitted wavefront. The second pulse is from the reflection of an object in the aperture. Source: [9].

There are two principal ways to reconstruct an image from a set of A-scans using the SAFT which both yield a similar result but differ in their computational cost. Each voxel eventually is assigned a certain value which corresponds to the intensity of that voxel in the final image. The intensity represents the echogenicity of the tissue at this particular position and is qualitatively proportional to the reflection coefficient.

For the **first method** one starts with the desired voxel that the value should be calculated for. A simple 2D example of the xy-plane from Figure 2.6 is shown in Figure 2.7. On the left is the aperture with the 3D volume shown as a 2D grid. The emitter $\vec{\chi}_e$ and the two receiving transducers $\vec{\chi}_r$ and $\vec{\chi}_{r+1}$ are on the outer shell of the aperture. On the right two A-scans are depicted. The upper A-scan shows the plot of the pressure of the configuration where emitter $\vec{\chi}_e$ transmits a wavefront and $\vec{\chi}_r$ measures the transition of the pressure. The plot below is for emitter $\vec{\chi}_e$ but this time with transducer r_{i+1} receiving. Considering how close $\vec{\chi}_r$ and $\vec{\chi}_{r+1}$ are placed to each other in this example the A-scan on the bottom shows an exaggerated negative time shift compared to the A-scan above.

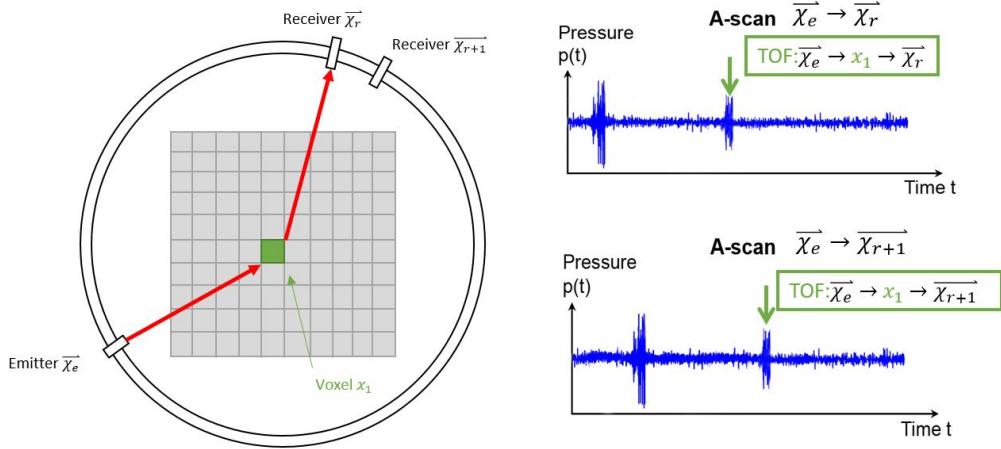


Figure 2.7: Basic principle of the SAFT. For one voxel every A-scan is analysed at a certain TOF. The final voxel value is the sum of every amplitude at the time in each corresponding A-scan. The procedure has to be repeated for each individual voxel.

For a known position of a voxel the TOF from each emitter to each receiving transducer can be calculated. For this basic example equation 2.1 shows how to calculate the TOF for the first voxel when neglecting the different speed of sound for the different types of tissue and assuming a constant SOS.

$$TOF_{1,ij} = \frac{dist_{\vec{\chi}_e, x_1} + dist_{x_1, \vec{\chi}_r}}{SOS_{water}} \quad (2.1)$$

$dist_{\vec{\chi}_e, x_1}$ corresponds to the euclidean distance of the emitter $\vec{\chi}_e$ to the voxel x_1 and $dist_{x_1, \vec{\chi}_r}$ to the euclidean distance from the voxel x_1 to the receiver $\vec{\chi}_r$. The speed of the sound wave is considered by SOS_{water} in the equation.

The TOF for the first emitter-receiver combination is shown as the green arrow in the upper A-scan in Figure 2.7. Here the TOF is directly located at the peak of the reflected wavefront. Since the A-scans have to be discretized the corresponding TOF may lay between two samples of the A-scan. To determine the appropriate value in the discrete set of samples either a linear 1D-interpolation between the two samples left and right of the TOF can be performed. Otherwise a nearest neighbour interpolation can yield a time sample. For this TOF the amplitude of the A-scan then is taken and added to the voxel value of voxel x_1 . In the same manner the TOF of the second receiver-emitter configuration is calculated and a sample in the A-scan can be found. This procedure is repeated for every emitter-receiver combination and every aperture rotation shift so that the final result is the voxel value only for voxel x_1 . To reconstruct the whole 3D image these steps have to be repeated for every voxel in the volume.

Equation 2.2 shows the calculations necessary to get the voxel value V_k for an arbitrary voxel k .

$$V_k = \sum_i^{N_e} \sum_j^{N_r} A(ToF_{k,ij}) = \sum_i^{N_e} \sum_j^{N_r} A \left(\underbrace{\frac{dist_{\vec{x}_e, \vec{x}_k} + dist_{\vec{x}_k, \vec{x}_r}}{SOS_{water}}}_{=TOF_{k,ij}} \right) \quad (2.2)$$

For N_r receiving and N_e emitting transducers the reconstructed value V of each voxel k consists of the sum of all pressure values at the TOF in the respective A-scans for all emitter-receiver combinations (N_e, N_r).

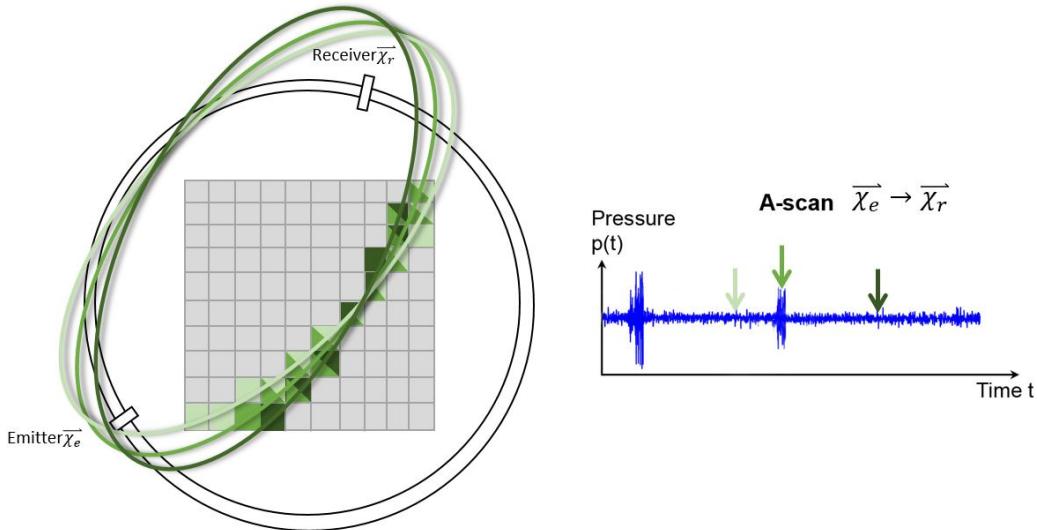


Figure 2.8: Alternative method to reconstruct the reflection image from the A-scans.

For the **second possibility** we do not start at a certain voxel but look at each individual A-scan. In Figure 2.8 the USCT aperture is shown this time with only one emitter-receiver configuration. On the right side the A-scan for this configuration is schematically plotted. In contrast to the first method we do not only pick one certain sample in the A-scan but a few. In this example three samples are chosen marked by the arrows in the different shades of green. For each of the three chosen TOFs there is one ellipse in Figure 2.8. Along those ellipses, which is cut by the ellipse, the amplitude value of the A-scan is added to the cut voxels. Bresenham's line drawing algorithm [13] may be used to decide which set of voxels approximates the ellipse best and therefore to which voxel the corresponding amplitude value is added. An example is shown in Figure 2.9.

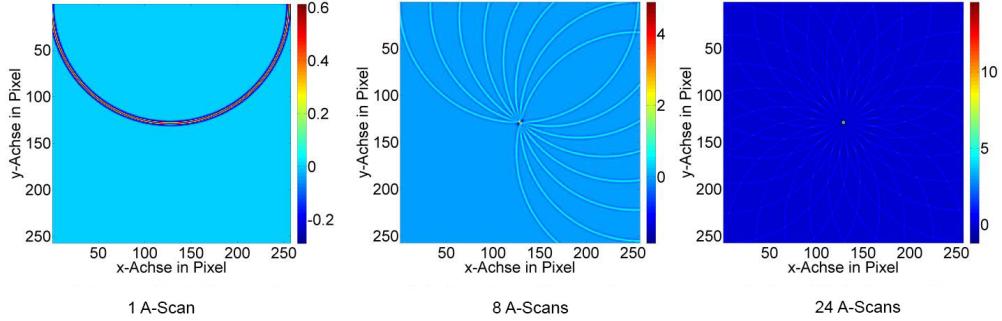


Figure 2.9: Superposition of multiple ellipses during the reconstruction of the final image. By averaging the voxel values of multiple ellipses each ellipse loses significance in the final image. Source: [7].

The example shows that the influence of each ellipse becomes less prominent with an increasing number of ellipses. Only at the point of scattering these ellipses superimpose and create a high amplitude that remains visible in the image.

Both methods have their advantages and disadvantages. However, during this thesis the first reconstruction algorithm was used. One of the main reasons for that is that all the calculations trivially be performed in parallel and therefore profit from the high number of threads of the GPU architecture. Furthermore, as was mentioned that the ellipse method makes multiple interpolations for each voxel in 3D necessary whereas the first method only interpolates in 1D between two sample values. By choosing the first method very expensive computation operations can be avoided.

2.1.3 Speed of sound correction

For the SAFT imaging there is no previous knowledge about the properties of the propagation speed in the medium. Until the SOS correction was introduced, for the SAFT imaging, the propagation speed of the ultrasound was approximated to match the propagation speed of a sound wave in water: $SOS_{path} \approx \bar{c}_{water}$ [9]. The SOS was assumed to be constant for the whole path for the calculation of the TOF in each A-scan in this case. This assumption neglects the fact that the medium is not homogeneously filled with water.

Since this is only a rough approximation of the actual SOS in the volume the assumption of a constant propagation speed led to a reduced contrast in the reconstructed image as the real location of the scattering was blurred during this process and small scatterers can not be resolved. To improve the contrast of the final image for each path the SOS is calculated. With Bresenham's line drawing algorithm [13] the appropriate voxels along the path from the emitter to the scattering voxel and further to the receiver are selected. For each of these N visited voxels the local speed $c(\vec{x}_k)$ is taken from the SOS-image from the preceding SOS measurement of the tissue.

The average speed of the sound propagation for that certain path \overline{SOS}_{path} then can be calculated with the harmonic mean in equation 2.3 [9]:

$$\overline{SOS}_{path} = \frac{N}{\sum_{k=1}^N \frac{1}{c(\vec{x}_k)}} \quad (2.3)$$

During the SAFT in equation 2.1 and 2.2 the SOS_{water} can be replaced by the more realistic \overline{SOS}_{path} for each individual propagation path.

2.2 Characterisation of reflections

During the acquisition of the data each emitting transducer emits a wavefront into the aperture which then interacts with the breast tissue in a distinct way. Therefore, it is important to take into consideration how the ultrasound wave interacts and scatters at the surface of the different tissue types and other objects in the aperture. The directional information of the reflections are lost during the classical approach of the SAFT [14]. As this thesis is focused on the tissue classification based on a back scattering model the reflection characteristics of the tissue can no longer be neglected. Figure 2.10 shows four different kinds of scattering and reflection depending on the structure of the surface and the direction of the incident ultrasound pulse.

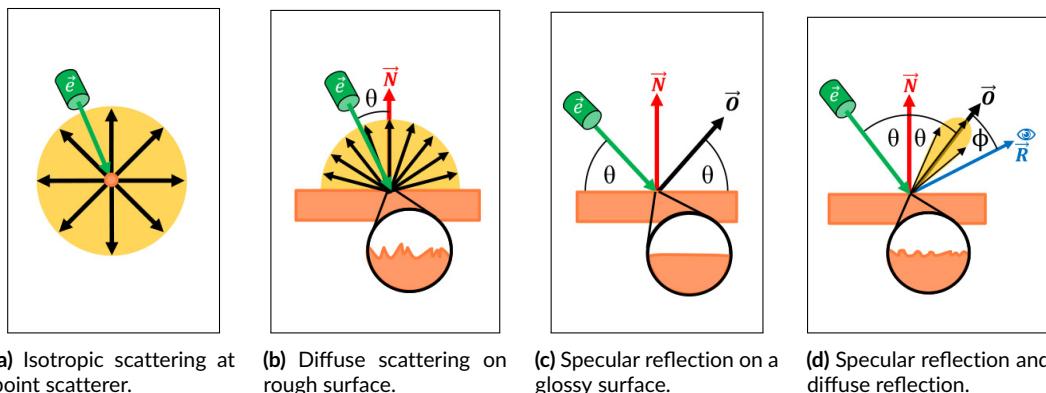


Figure 2.10: Reflection characteristics of different surfaces and angle of incidence. Source: [14]

Figure 2.10a shows the principle of isotropic scattering at a point scatterer. Transducer \vec{e} emits a soundwave into the volume. When the sound wave reaches the point of scattering it will be reflected homogeneously into every direction with an isotropic distribution of intensity. This emitted spherical wave has a constant energy distribution in each direction. This is a theoretical principle of a reflection which the SAFT is based on since it neglects all directional information of the received sound wave.

A better scattering model can be seen in figure 2.10b. The diffuse scattering occurs on a rough and dull surface where the incoming energy of the ultrasound wave is equally scattered in every direction above the surface of the material (Lambertian reflectance). The surface of the material then appears to have the same radiance from every angle. The Lambert's Cosine Law states that the angle of the inciting wavefront has an influence on the radiated intensity [15]. The smaller the angle θ between the incoming wavefront and the surface normal \vec{N} , the larger the scattered intensity from the surface. In other words: the diffuse reflection is proportional to the amount of energy in form of ultrasound waves that hits the surface per unit area [16], which is represented by equation 2.4:

$$I_{\text{diffuse}} \propto \cos(\theta) \quad (2.4)$$

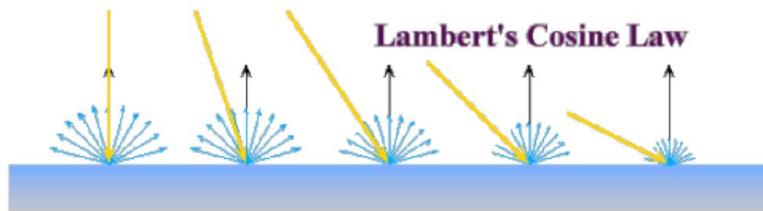


Figure 2.11: Representation of the Lambert's Cosine Law. The scattered intensity from the surface is proportional to the cosine of the angle between the incoming wavefront and the surface normal. Source: [16].

The third reflection characteristic is called specular reflection and can be seen in figure 2.10c. Whereas in the case of the diffuse scattering the energy density of the scattered wave is evenly distributed above the surface facing the emitter, the specular reflection has a concentrated direction of emission. For glossy and even surfaces the incoming wave is reflected on the surface ideally whereas the entry angle θ is equal to the exit angle θ . The main reflection direction is described by vector $\vec{\sigma}$. For the ideal case of specular reflection the angle of the incident soundwave equals the outgoing soundwave. Therefore the receiver normal has to be directly within the angle of the emitter normal to the surface. Otherwise the receiver would not detect any signal.

Figure 2.10d shows a combination of specular reflection and diffuse scattering. This model is appropriate for materials with a surface structure somewhere in-between rough and glossy. The emitted intensity is a mixture between the directly reflected energy based on the specular reflection and the scattered energy of the diffuse scattering model. This results in a cone shaped beam with an opening angle based on the scattering factor of the material. Again, the angle between the incident wavefront and the surfaces normal determines the intensity of the reflected wave. Now also applies that the angle Φ between the receiver vector \vec{R} to the main reflection direction $\vec{\sigma}$ affects the received intensity as well. The smaller Φ becomes the greater is the measured intensity.

2.3 Graphic processing unit (GPU)

For the reconstruction of the image from the A-scans a sequence of calculations has to be performed which are highly parallelisable. In this section the benefits of utilizing graphic processing units (GPUs) to manage these calculations shall be addressed.

A main requirement for parallel calculations is that they can be performed independently from each other. If there are certain dependencies between multiple simultaneous calculations so that one thread might have to wait on the intermediate result of the calculations of another thread then the whole process often becomes inefficient and unstable. In later chapters it will become clear that most of the calculations that are necessary for the reconstruction can be performed independently of each other. One example is the SAFT during which for each voxel a certain value V_k is determined. This process can be performed independently from the other voxels. Beforehand, it was also mentioned that these processes of computation have to be repeated for each individual voxel. For a typical relevant volume of 1024x1024x1024 voxels this process can be parallelised for each individual voxel and profits from the GPUs architecture.

These kind of operations fall within the scope of so-called single instruction, multiple data (SIMD)-operations. These SIMD-operations are qualified by repeating the same or at least very similar operation on a changing set of data. Particularly, for this scenario GPU are fitted with a high number of computation cores and high bandwidth memory buses.

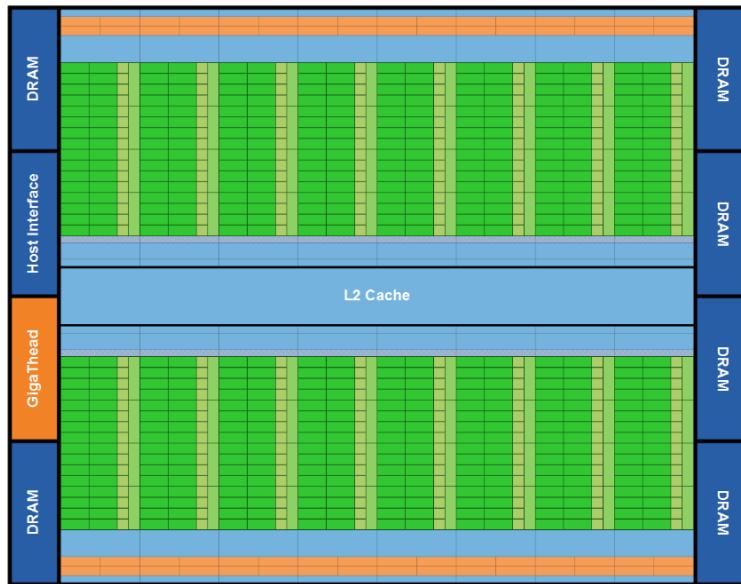


Figure 2.12: Example of a Fermi GPU with 16 streaming modules(SM). Each SM is connected to the central Level 2 Cache which connects each core with each other. For each SM there are the 32 execution units coloured in green [17].

The current generation of RTX NVIDIA GPUs has up to 82 so-called streaming modules (SMs) with a total of up to 10496 arithmetic logic units (ALUs) [18]. These ALUs are the unit of a processor which handles all operations which are necessary for the calculations. Simple arithmetic operations like for example additions and multiplications as well as logical operations like negation or AND-Conjunctions are managed by the ALU. The schematic of a basic GPU architecture can be seen in Figure 2.12. The schematic shows 16 SMs with 32 so-called CUDA-Cores each. The CUDA-Core represent the execution entities of the GPU. Each SM is connected to the Level 2 (L2) Cache which connects each SM to the other. Furthermore, around those SMs there are units for storing larger amounts of data. The so-called DRAM, often also GDDR, is the random access memory of the GPU. If enough GPUs are available it is possible to process large reconstruction volumes separately but in parallel and therefore minimise the memory access overhead when transferring data from the host to the device or vice versa.

In comparison a typical Intel i9 desktop central processing unit (CPU) in the 10th generation has 10 physical and 20 logical processing units to work with [19]. Besides the much higher number of processing units of the GPU another big topic that is often overlooked is the memory bandwidth of the system. Of course, the number of ALUs have a big impact in how fast each set of calculations can be finished. After finishing with the calculation the results have to be moved to the memory to wait for further processing. Often the bottle neck can be found in the data connection of the ALUs to the host memory where ultimately every result is stored. Whereas the aforementioned Intel CPU has a memory bandwidth of 94 GB/s the NVIDIA GPU profits of its 936 GB/s. Furthermore, what makes GPUs so attractive for the reconstruction process of USCT-images is the hardware implemented texture unit. During the SAFT there is the need for interpolating between two or more samples to find a feasible value for the A-scan. This operation can be done either in software or as a specialised hardware implementation which can process the data at a much higher speed.

2.3.1 CUDA-programming

Besides having very powerful hardware at hand it is equally important to manage the data of the reconstruction algorithm as efficiently as possible. With the introduction of the CUDA-programming technique NVIDIA made GPU-programming much more accessible. While the majority of code is written in C respectively C++ the actual core of the reconstruction is written in CUDA. The main objective of CUDA is to provide the programmer with a simple interface for her or his program components and get that particular software component to run on the GPU. Some operations that are necessary before the code can be executed by the GPU are for example the allocation of enough memory on the GPU. Furthermore, the data has to be copied from the host computer to the CUDA-Device. The GPU itself has no access to the RAM of the host machine. After that the CUDA-Kernel can be called. This is the central part of software that processes the raw data (in this case the A-scans) and yields the results of the calculation.

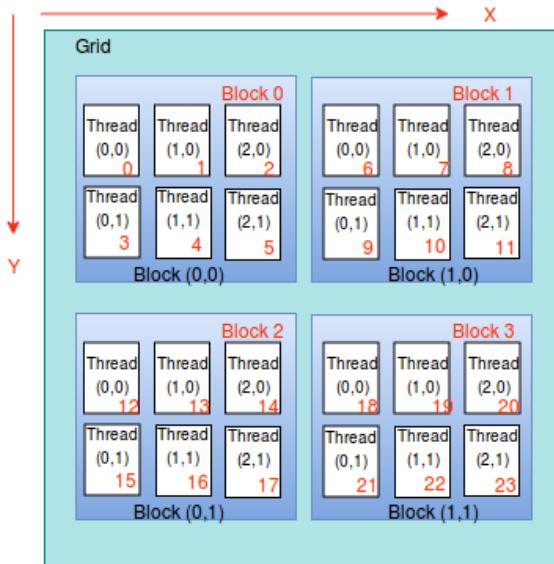


Figure 2.13: The organisation of threads when calling the kernel. Each thread is organised in blocks. These blocks again are substructures which are organised in grids [20].

When a CUDA-Kernel is called besides the main function there are two parameters that have to be set. These manage the indexing behaviour of the input data: the number of blocks per grid and the number of threads per block.

```
kernelFunction<<<numBlocks, threadsPerBlock>>>
```

The indexing of a CUDA-Call can be seen in Figure 2.13. The parallel execution of a certain code segment leads to a new thread. Each thread serially numbered by the red numbers from 1 to 23 is organised in a 2D Block-Grid-Structure. The basic idea is that a certain number of threads can be organised into blocks. How many threads are summarised as a block can be managed by

```
threadsPerBlock
```

So the first 6 threads (0 - 5) are organised as the first Block 0. Each thread gets an internal ID the so-called

```
threadIdx
```

The `threadIdx` is an up to 3D indexing-array to classify each thread running in each execution block. This can also be seen in Figure 2.13 where each of the threads is indexed by a 2D vector.

A number of blocks then again can be summarised in a so-called grid. That is what the second parameter for the Kernel-Call is for: With

```
numBlocks
```

Similar to the threads each block has a multi dimensional ID the so-called `blockIdx`

An example is given in Figure 2.14:



Figure 2.14: Example of how multiple, simultaneously executed threads are assigned to their `blockIdx` and `threadIdx` [21].

In this example there are each 8 threads per block. Their `threadIdx.x` are serially numbered from 0 to 7. The first block is assigned `blockIdx.x = 0`. The `.x` indicates that in this case the `threadIdx` as well as the `blockIdx` are one dimensional and only have elements in x-direction. This process of enumeration is repeated for all 32 threads so that every thread can be accessed by means of block-ID and thread-ID. Theoretically, these four blocks now can be regarded as a grid. A maximum of 1024 threads can be summarised as a block. A grid may contain up to 65535 blocks. Each block per grid then contains the same number of threads. The scheduling of all the threads onto the CUDA Cores is done by the CUDA Engine itself. Furthermore, the parameters of the kernel call set how often the kernel function is called in total. In the kernel function the right set of data has to be picked depending on the `threadId` and the `blockId` so that the thread processes the right data.

Methods

3.1 Extension of the SAFT information dimensionality

One of the main objectives of this thesis is the introduction of a fourth imaging modality for the reconstruction algorithm of 3D USCT images. This modality can be used to for a classification of different tissue types by analysing the scattering characteristics of each tissue sample.

As it was mentioned in the motivation in Section 1.1 the three dimensional volume of the image is extended to differentiate the directional information of the data over two additional dimensions. The three dimensional spacial information is extended by a two dimensional angular information of the emitter and receiver configuration. This approach results in a five dimensional image volume.

The idea shall be explained by an example with Rubik's cubes: The typical 3D reconstructed image shall be represented by a white Rubik's cube as seen in Figure 3.1:



Figure 3.1: Standard SAFT: three dimensional spacial representation of the image as a Rubik's cube.

A SAFT-image has three dimensions with the corresponding voxels which are here represented by the blocks of the Rubik's cube in Figure 3.1. Each block has a voxel value V_k which results from the superposition of each A-scan-sample from the SAFT reconstruction.

This is the standard case where no directional information is available. From this SAFT representation the emitter and receiver information can not be differentiated.

The introduction of the 4th dimension helps with the preservation of directional information. Basically, the three dimensional volume of the image is split into as many parts as there are directional vectors. The directional vectors basically are a discretised form of the directional information around each voxel which helps to quantify the direction from where the signal originates or goes to. The example for four directional vectors can be seen in Figure 3.2:

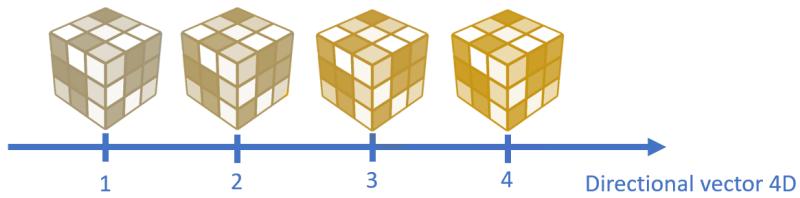


Figure 3.2: The four dimensional image for a total of four directional vectors. For each directional discretisation an additional three dimensional volume is created, hence the four Rubik's cubes. The directional information is represented as the saturation of the different cubes.

The A-scan is assigned to a certain voxel in one of the four cubes. Each cube belongs to one receiver if the receiver information is stored in the 4th dimension. If the emitter information is regarded, each cube contains the data of one emitter. Information about emitter and receiver can not be stored simultaneously. The schematic of the configuration of emitters and receivers can be seen in Figure 3.3:

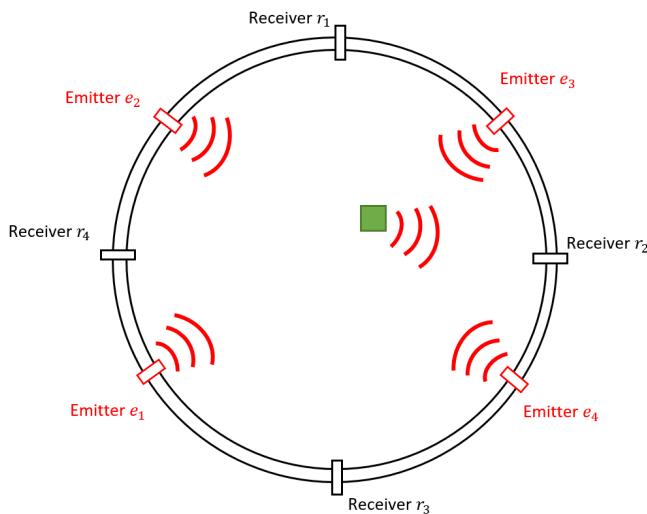


Figure 3.3: Example of imaging aperture with four receivers and two emitters.

In Figure 3.3 a simple example for four receivers and four emitters is given. The green square in the aperture represents an arbitrary voxel in a tissue sample which shall be examined. What directional information is stored in each dimension of the four Rubik's cubes depends on what configuration is of interest. One possibility would be to assign each receiver direction to a Rubik's cube. In that case the information of the emitters would be lost. In this example the emitter e_1 could be the emitting transducer and receiver r_2 the receiver. This would result in an A-scan which can be processed with the SAFT as was shown in section 2.1.2. The resulting voxel value V_k of the SAFT would be assigned to the 2nd Rubik's cube in Figure 3.2. If this procedure is repeated for every A-scan there is, each voxel value would be assigned to one of the Rubik's cubes that belong to the corresponding receiver index. The information of which emitter belongs to which voxel value is lost i.e. the directional information is averaged for all emitting directions. The summation of all Rubik's cubes along this new 4th dimension would result in the three dimensional SAFT image that has no directional information and was shown in Figure 3.1.

This approach could also be used to analyse the emitter configuration which would lead to a set of Rubik's cubes that are assigned to the emitter direction. In this case the information of the receivers would be lost.

A 5th dimension can be used to keep both information about the emitter as well as the receiver. For this, the four Rubik's cubes that represent the fourth dimension are repeated into the 5th dimension as many times as there are directional vectors. For four directional vectors this leads to $4 \times 4 = 16$ total Rubik's cubes. An example of this is shown in the following figure:

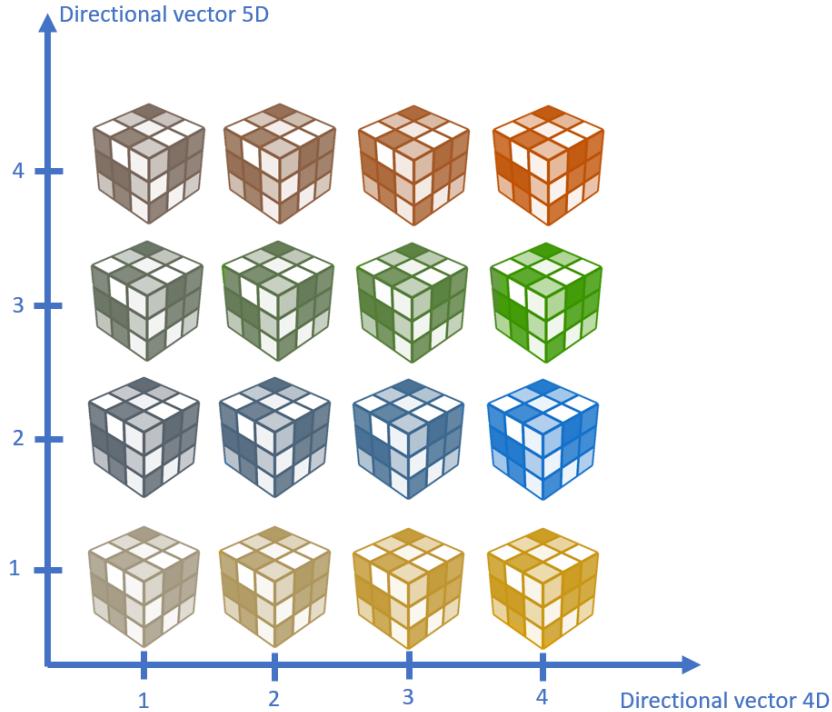


Figure 3.4: Example of a five dimensional set of Rubik's cubes. The information in the 4th dimension is represented by the saturation and represents the receiver direction. The emitter information is kept in the 5th dimension which is represented by the different colours of the cubes.

In the case of four directional vectors (i.e. here four receivers or four emitters) the resulting five dimensional volume consists of 16 Rubik's cubes each containing the information in which direction the emitter lays as well as in which direction the receiving transducer was. For the example in Figure 3.3 one possible configuration could be that emitter e_2 sends an ultrasound wave into the aperture. This pulse scatters at the location of the green voxel and is reflected to receiver r_1 . The voxel value V_k for this configuration would be assigned to the Rubik's cube at the coordinates (1, 2) in Figure 3.4 and would be entered at the corresponding voxel position of that particular Rubik's cube. With this method now an arbitrary resolution of the directional discretisation can be handled. From this 5D volume the reflection characteristics of different tissue types can be plotted and compared.

One downside of this partitioning of the A-scans into many sub-volumes is the loss of contrast in the individual images. Each Rubik's cube contains only a part of the initial 3D SAFT image. Since the number of A-scans is constant each new volume receives only a fraction of the available A-scans. The overall contrast therefore is distributed between the different Rubik's cubes and therefore lower for one cube alone.

A challenge of the additional dimension is the exponential growth of the data that has to be processed. A typical 3D volume of $250 \times 250 \times 250$ voxels contains 15.6×10^6 elements. Each element has a voxel value that is stored as a double value of 8 byte. For the 3D volume

alone 119 MB of data have to be stored. The 5D approach increases this amount of data to 1.9 GB for only 4 directional vectors.

3.2 Discretisation of directional information

The first step of the preservation of directional information during the reconstruction is the generation of a suitable set of directional vectors which divide each voxel volume as evenly as possible. Theoretically, a continuous approach for the directional information could be pursued. This would lead to set of directional vectors for every voxel to every receiver and every emitter of the USCT. For this case a constant set of directional vectors would not be feasible since every voxel position affects the vectors from the voxel to the corresponding receivers and emitters. Therefore, an image volume of $250 \times 250 \times 250$ voxels with 157 TAS and each TAS containing 4 emitters and 9 receivers would lead to 1.387×10^{12} directional vectors, if 10 aperture positions are acquired. Each vector is classified by three double values which would lead to 3026.4 terabytes of data that has to be stored solely for the vector orientations. If the actual image data would be assigned to this hypothetical set of directional vectors the storage requirements would increase manifold. Thus, a trade-off between the resolution of the discretisation and the storage requirements has to be found. To achieve that, the directional information for each voxel is discretised into a set of directional vectors. The directional vectors then are placed into the centre of each voxel facing outwards. During the reconstruction for every directional vector an individual volume is created. This was explained in section 3.1. Theoretically, the directional vectors do not have to be distributed equally. If a certain direction requires a higher resolution of the segmentation it would be no problem to either add more vectors to that particular direction or shift some existing vectors from a less relevant direction to that required direction. In the following it was aspired to distribute the vectors as evenly as possible since no information about the relevance of one certain direction over another are available during the time of writing. Section 3.2.1 explains how the properties of platonic solids can be used to generate a set of either 12 or 20 uniformly distributed vectors. The limitation of the amount of vectors which arises with the usage of platonic solids is tackled by the method of arbitrary segmentation in section 3.2.2.

3.2.1 Platonic solids

Since the goal of the segmentation is to divide the measurement volume of each voxel as evenly as possible the intermediate angle between two neighbouring segmentation vectors should be equal. Platonic solids are one possibility to get a set of vectors which fulfil this requirement and was presented by Patrick Hücker [7].

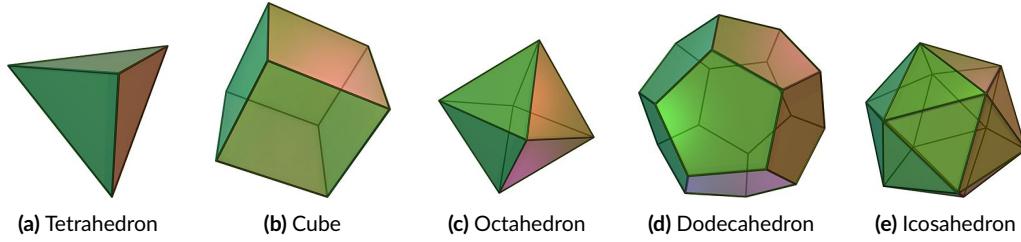


Figure 3.5: The five platonic solids. They all have equally sized faces. Source: [22]

A key feature of platonic solids are the equally sized faces and therefore the evenly distributed face normals if placed in the centre of each face. There are five different platonic solids (shown in Fig. 3.5): The tetrahedron with four faces, the cube with six faces, the octahedron with eight faces, the dodecahedron with 12 faces and the icosahedron with 20 faces. From these five platonic solids the dodecahedron as well as the icosahedron have previously been used to generate suitable segmentation vectors. Theoretically, the remaining geometries provide a suitable set of evenly distributed normals for the segmentation of the volume as well. Since they only have eight or less faces the comparably low resolution of the segmentation of the volume would make an analysis of the direction of propagation rather difficult.

Figure 3.6a shows the implementation of the dodecahedron in MATLAB. Each face normal as well as the corresponding index is plotted in the centre of each face.

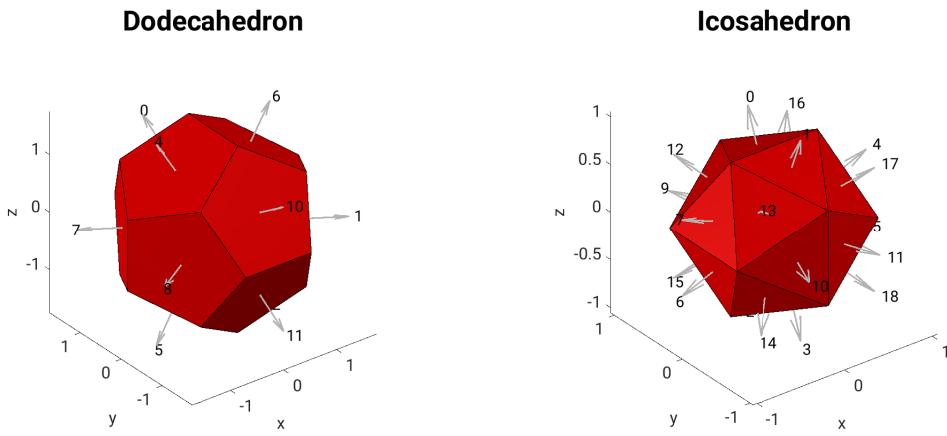


Figure 3.6: Both platonic solids realised in MATLAB. In the centre of each face the normal vector is plotted with its corresponding index.

3.2.2 Arbitrary directional segmentation

The reflection characteristic of a medium depends on the angle under which an incoming ultrasound wave is scattered and reflected. To distinguish between very narrow angles for specular reflections and the broader angles in the diffuse case the directional information has to be given in a sufficient resolution. If the resolution of the platonic solids is not sufficient to distinguish between specular and diffuse reflectivity there has to be a way to increase it arbitrarily. This is why one fundamental part of this thesis was the augmentation of the method for the generation of suitable directional vectors to discretise the directional information arbitrarily in contrast to the fixed number of normals that were yielded by the platonic geometries. To be able to increase the density of the segmentation vectors it is necessary to create an arbitrary amount of them. The challenging part is the alignment of the directional vectors so that every neighbouring vector has a similar angle to its other neighbours. The problem of generating a set of uniformly distributed vectors is known as the Thomson Problem [23]. It was formulated as the problem of the minimisation of the electro-static potential energy of N equally charged particles located on the surface of a unit sphere. The particles were all equally charged and therefore repel each other following Coulomb's Law. The equation for the Coulomb potential is shown in Equation 3.1 for two charged particles i and j :

$$\Phi(r_{i,j}) = \underbrace{\frac{1}{4\pi\epsilon_0}}_{=k_e} \frac{q_i q_j}{r^2} = k_e \frac{q_i q_j}{|r_{i,j}|^2} \quad (3.1)$$

With $|r_{i,j}|$ being the Euclidean distance between two charged particles with the charges q_i and q_j . k_e is the electrostatic constant. Assuming that both particles are equally charged ($q_i = q_j = 1$) the equation for the electrostatic potential energy can be simplified:

$$\Phi(r_{i,j}) = \frac{k_e}{|r_{i,j}|^2} \quad (3.2)$$

The combination of a set of N charged particles results in the following potential:

$$\Phi(r_{i,j}) = k_e \sum_{i=1}^N \sum_{i \neq j} \frac{1}{|r_i - r_j|^2} \quad (3.3)$$

The optimisation problem can be expressed as a penalty function with the so-called Reisz s-energy $\epsilon_s(r_{i,j})$ as seen in Equation 3.4. The goal is the minimisation of the Reisz s-energy by the means of maximising of the distance between the particles [24]:

$$\min \epsilon_s(r_{i,j}) = \min \sum_{i=1}^N \sum_{i \neq j} \frac{1}{|r_i - r_j|^s} \quad (3.4)$$

In Equation 3.4 s is the the Reisz s-energy parameter which defines the penalty for the smallest distance between two points. This means that with higher s the optimisation algorithm will try

to the maximise the overall distance between the points to reach a better solution compared to lower s values. This comes with the downside of increased computational expense. Furthermore, k_e can be neglected since it is constant and does not have an influence on the solution of the optimisation problem. The actual solution for this optimisation problem can be found by applying numerical approaches like the line search algorithm.

For the calculation of the final point configuration the implementation from [25] was used. The result is an approximately uniform triangular configuration of points on a sphere for a set of N electrically charged particles. The algorithm itself does not make use of a line search algorithm. Instead it distributes the particles in a random manner and uses thresholds to decide if a new configuration yields a better solution of the optimisation problem than the one before. Then new constraints are defined and the particles are again distributed randomly until they fall below a threshold. This is repeated until a given quality of the optimisation is reached. Figure 3.7 shows four examples for the resulting set of N equally distributed points on the sphere.

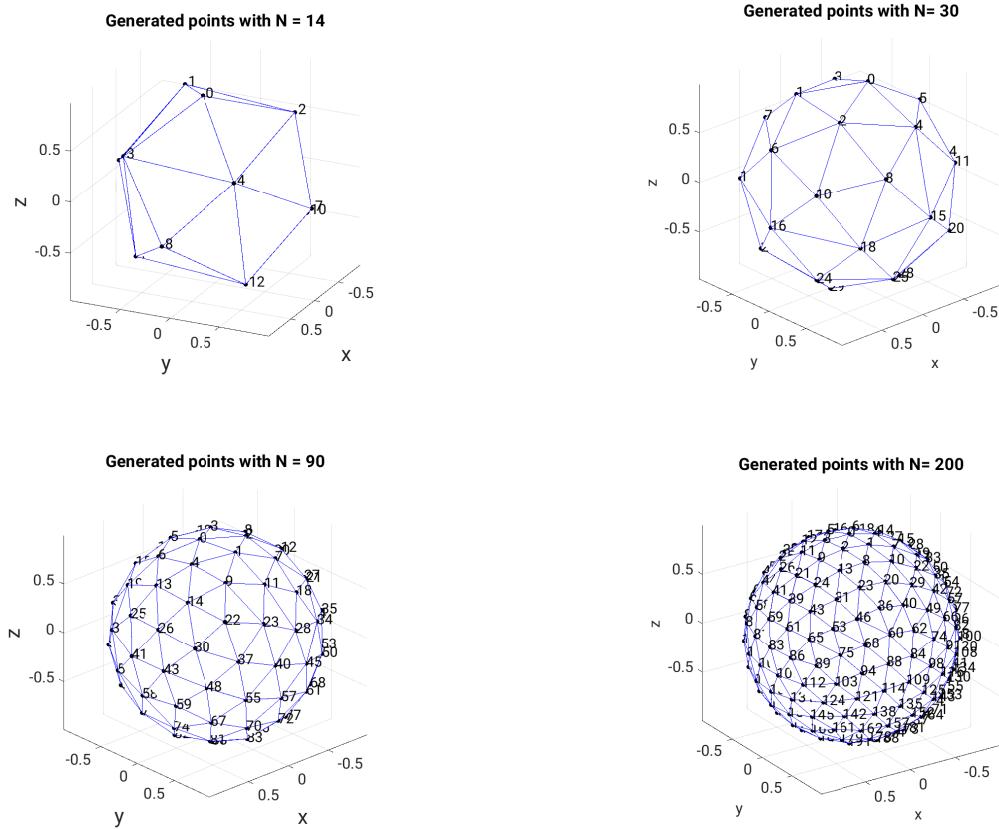


Figure 3.7: With the arbitrary directional segmentation approach it is possible to generate an arbitrary set of equally spaced points on the surface of a sphere which then can be used to create the vectors to segment the volume.

From this triangular tessellation it is possible to derive a set of vectors which segment the measurement volume. The start point of each vector is chosen as the centre of a voxel. The vectors then are constructed to the points of the tessellation. The indexes on the points represent the order in which they were generated. A physical context can not be derived from the enumeration. This is exemplified for $N = 14$ vectors in Figure 3.8:

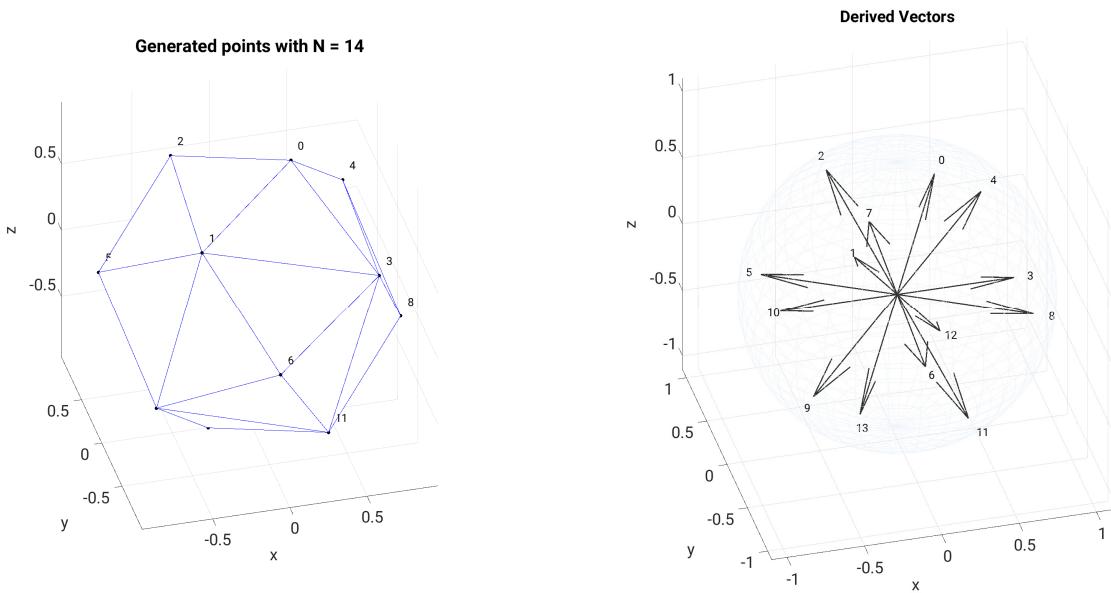


Figure 3.8: Left: A distribution of $N = 14$ points which equals Figure 3.7 in the top left. Right: The derived origin vectors for those 14 points.

Since the algorithm primarily assigns each point to a random position the results would be non-reproducible. Therefore, the seed of the random number generator was set to one for each iteration so that the resulting set of points would lead to the generation of comparable vectors.

The numerical solutions of the Thomson Problem with different numbers of particles comes to the conclusion that the minimisation of the energy results in the particles taking the form of known polyhedra. Depending on the number of charged elements N different polyhedra can be expected. For $N = 4$ the minimisation of the energy results in the alignment of the electrically charged particles in the form of a tetraeder. For example for $N = 6$ they will form an octaeder and with $N = 12$ they will approximate an icosaeder. These forms are part of the platonic solids that were introduced in chapter 3.2.1. Thus, it does not make sense to generate a uniform distribution for $N = 12$ and $N = 20$ particles as we can reach these results

in a much more efficient way by utilising the geometry of the platonic solids. Nevertheless, the implementation allows any number of $N > 2$ particles to find an optimal solution even if the platonic solids provide a more efficient alternative for certain numbers of particles.

3.3 Comparison vectors

The SAFT reconstructions results in a value V_k for a particular voxel. The discretisation of the directional information now allows for the voxel value to be assigned to a certain directional vector. Each volume of the final image for each directional vector will be assigned only to those voxel values that belong to that directional index. To assign a certain voxel value V_k to each of the directional vectors' indices a comparison vector has to be defined for that A-scan. For this thesis mainly the comparison vectors comprise the vector from the position of the voxel which is analysed in that certain step to the position of the receiving transducer:

$$\vec{V}_{comp, vox_rec} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{rec} - \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{vox} \quad (3.5)$$

and the vector from the voxel position to the emitter:

$$\vec{V}_{comp, vox_emit} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{emit} - \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{vox} \quad (3.6)$$

It is essential to normalise the comparison vectors as well as to define them starting at the voxel position. In case of defining for example the directional vector from equation 3.6 from the emitter to the voxel the reconstructed image would be rotated 180° since the algebraic sign of the orthogonality is switched. In previous approaches only one directional vector was used. The reason for that was the assumption of a specular reflectivity model of the tissue which was mentioned in section 2.2. The usage of two different directional vectors for the emitter-voxel relation as well as for the receiver-voxel relation makes it possible to actually analyse the reflection characteristics of the tissue. For the receiver and emitter a new dimension of the reconstructed image is introduced which allows for the sampling of the reflection characteristic of each voxel and quantify the specular and diffuse parts in the reflection. The result is a five dimensional image with all the directional information stored in one of the arrays as it was explained in section 3.1. This potentially leads to a higher diagnostic value of the tissue classification than the previous case.

3.4 General procedure for calculating directional information

Figure 3.9 shows the general procedure of reconstructing an USCT image with directional information. The flowchart shows the algorithmic description of the assignment process that was exemplary described in section 3.1 with the Rubik's cubes. The stages in grey originate from the previous implementations whereas the stages in colour contain mostly new or extended approaches for the reconstruction.

The old implementation lacked the possibility to generate 5D images. Memory limitations were one of the main reasons that the 5th dimension could not be regarded. This limitation was overcome with the new implementation of the reconstruction.

The first stage of the flow chart is the **start of the reconstruction**. This step summarises all the preprocessing requirements of the SAFT for example the quality evaluation of each A-scan and the identification of suitable receiver-emitter-configurations. After the preprocessing the measurement volume is segmented by **generating a set of directional vectors**. This procedure of setting up the directional vectors is explained in section 3.2. At this point theoretically any vector in the voxel could be assigned to a directional vector.

Then, with the directional vectors and a suitable set of A-scans it is possible to start with the assignment loop. The loop begins with the first A-scan (`AScan[0]`) and the first directional vectors for the 4th and the 5th dimension (`directional vector4D[0]` and `directional vector5D[0]`).

The **SAFT** section of the flow chart considers each of the A-scans and calculates a voxel value V_k as it was explained in section 2.1.2. During this process also the speed of sound correction explained in section 2.1.3 is taken into account, which leads to an overall increased contrast and resolution of the final image.

The stage following the SAFT covers the process of deciding on a so-called **comparison vector** for each iteration of the loop. Section 3.3 gives more details on that step.

The central part of the reconstruction is implemented in the 5th stage. In this stage the directional information for each A-scan is analysed and the voxel value V_k assigned to the correct voxel in the corresponding dimensions. The previous implementation was heavily constricted by memory requirements and its execution time. A new method is introduced in section which overcomes these restrictions and allows for the processing of the directional information in 5D.

After the assignment of the voxel value V_k to the suitable directional index the process of calculating a voxel value V_k and determining a comparison vector starts anew with the next A-scan. Since all the calculations have to be repeated for every directional vector, once arrived at the last A-scan the next directional vector for the 4th dimension is loaded and the whole process repeats itself. After all `directional vector4D` were processed the outer loop selects the next `directional vector5D` and the algorithm starts from the SAFT stage again. Since there are multiple repeated executions of the same functions in this algorithm it is important to optimise each calculation step as much as possible.

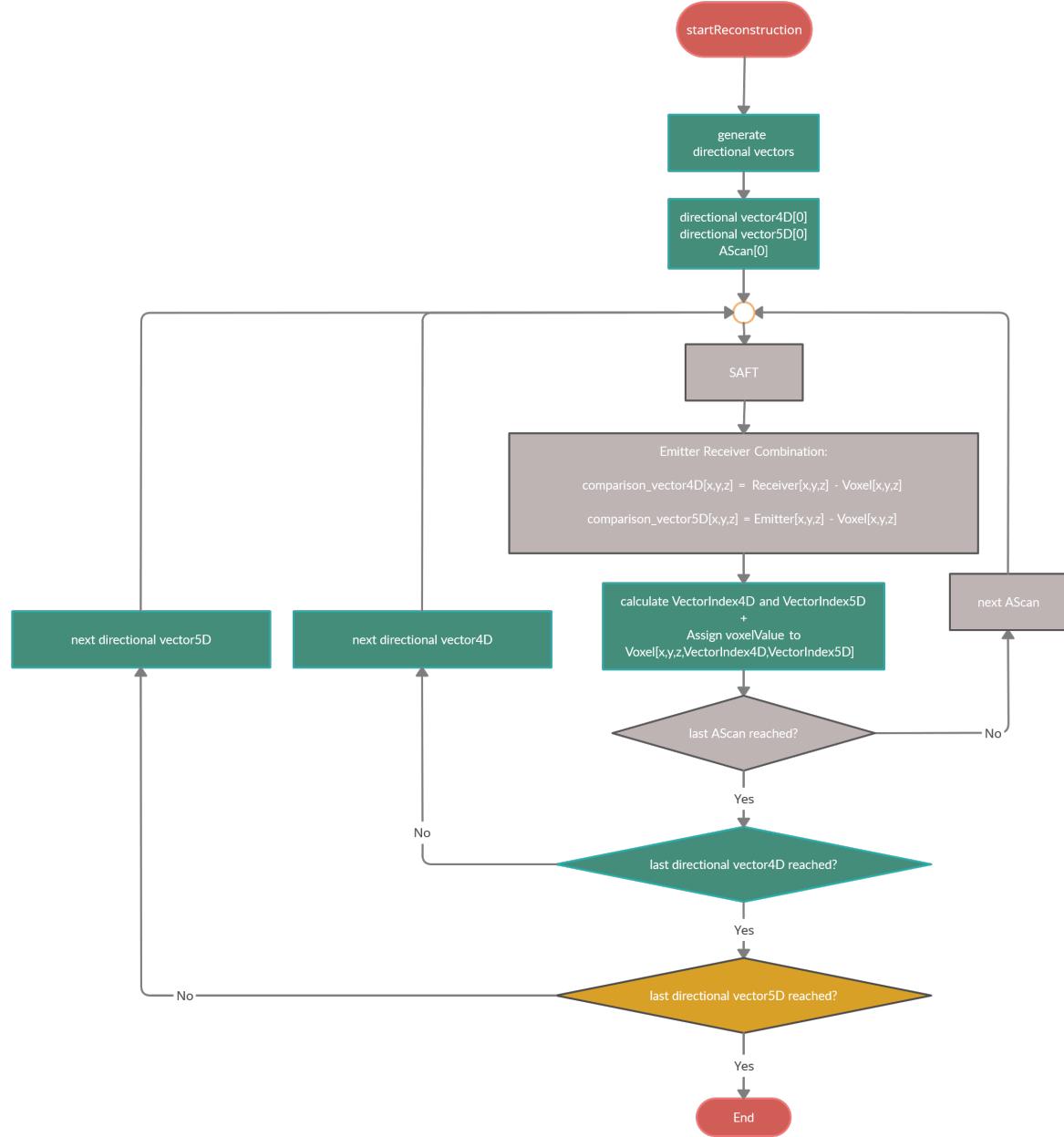


Figure 3.9: Flow chart of the algorithm for the reconstruction of an USCT image while preserving directional information in the data.

3.5 Assignment to the directional dimensions

With the methods explained in section 3.2 we receive a set of directional vectors which divide our measurement volume in an uniform manner. Furthermore, with the methods from section 3.3 one obtains two comparison vectors for each A-scan in the five dimensional case. The SAFT reconstruction results in a voxel value V_k that results from this emitter-receiver-configuration. The next step is the identification of a feasible directional vector for each A-scan and the assignment of the voxel value V_k to this vectors index. Two different methods will be presented: the angle sorting method which was adapted from Patrick Hücker [7] and a novel approach, the orthogonality threshold. Both methods will be called for every A-scan, for every possible directional vector in 4D and also for every directional vector in 5D. The assignment of the directional index to the comparison vector takes place in the dotted stage of the reconstruction algorithm which is shown in Figure 3.9. First it shall be shown that the orthogonality is a good metric to find the closest angle between multiple vectors and how to use it in both methods to assign a directional index to the comparison vectors.

3.5.1 Orthogonality

To assign each A-scan to a certain direction the orthogonality provides a good metric since it is independent of the azimuthal rotation between two vectors \vec{a} and \vec{b} . To get a conclusive result concerning the angular relation of two vectors one requirement of this method is that every vector has to be normalised. If this requirement is met one can arrive at the definition of perpendicularity:

Two vectors \vec{a} and \vec{b} are considered orthogonal to each other if the following assumption is fulfilled:

$$\vec{a} \perp \vec{b} \Leftrightarrow \vec{a} \cdot \vec{b} = 0 \quad (3.7)$$

Since this case only covers the case of an exact right angle between both vectors it is important to have a look at the distribution of the orthogonality between the vectors \vec{a} and \vec{b} for different angles. An example is shown in Figure 3.10. Vector $\vec{a} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ is represented by the green arrow on the left side. It serves as a constant reference vector. On the right side there are shown three possible vectors \vec{b} .

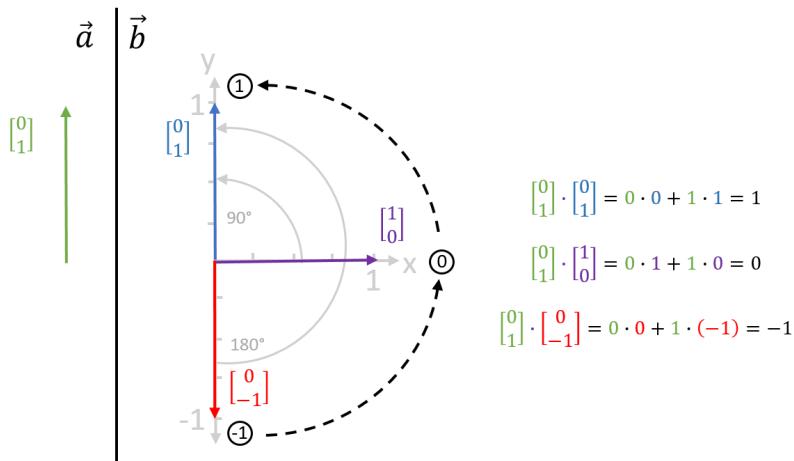


Figure 3.10: Example for the calculation of the orthogonality of one static reference vector \vec{a} on the left side and three possible candidates for \vec{b} on the right. The calculation shows the transition of the orthogonality from its minimum of -1 to the maximum of 1 .

The calculations on the right side of Figure 3.10 show the orthogonality between each combination of \vec{a} and \vec{b} . For example vector \vec{a} and the purple vector \vec{b} are perpendicular to each other. The calculations show that the orthogonality between both is zero and therefore fulfil the requirement of Equation 3.7. The red vector \vec{b} and the green comparison vector \vec{a} comprise an angle of 180° and therefore both vectors are pointing in the exact opposite direction. This results in an orthogonality of minus one. If the vector \vec{b} is pointing in the same direction as vector \vec{a} they are parallel and the orthogonality between both is one. The dotted semicircle shows the transition of the orthogonality from its minimum to its maximum. In this interval the orthogonality monotonically increases non-linearly from -1 to 1 . The reason for the non-linear behaviour is the correlation of the scalar product with the cosine function. The scalar product of two vectors corresponds to the cosine of the angle between those vectors. For a right angle (90°) the cosine becomes zero and for parallel case (0°) the cosine results in one. An example of the non-linearity of the orthogonality and its relation to the cosine function is given in Figure 3.11:

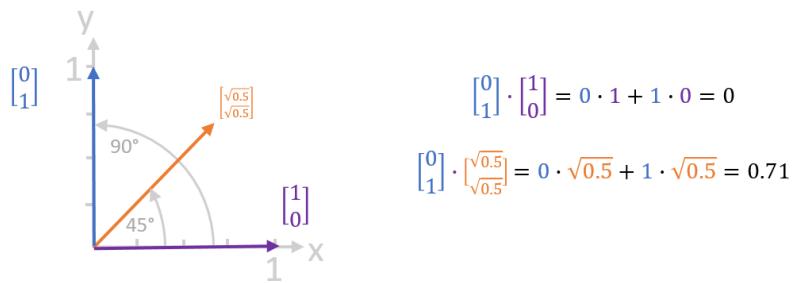


Figure 3.11: Example of the non-linearity of the orthogonality in the interval $[-1, 1]$.

The orthogonality between the blue vector $\begin{bmatrix} 0 \\ 1 \end{bmatrix}$ and the purple vector $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$ is zero since both are perpendicular. The normalised bisector in between is shown by the orange vector. The orthogonality between the blue vector and the bisector is approximately 0.71.

An alternative derivation of the method to identify the direction vector is given in A.1.

3.5.2 Angle sorting method

The first method for assigning a directional index to a certain comparison vector will be explained in this chapter. It is an adaption of the angle sorting method from Patrick Hücker [7]. It was extended to the 5th dimension. The assignment process takes place in the central stage of the flowchart shown in Figure 3.9.

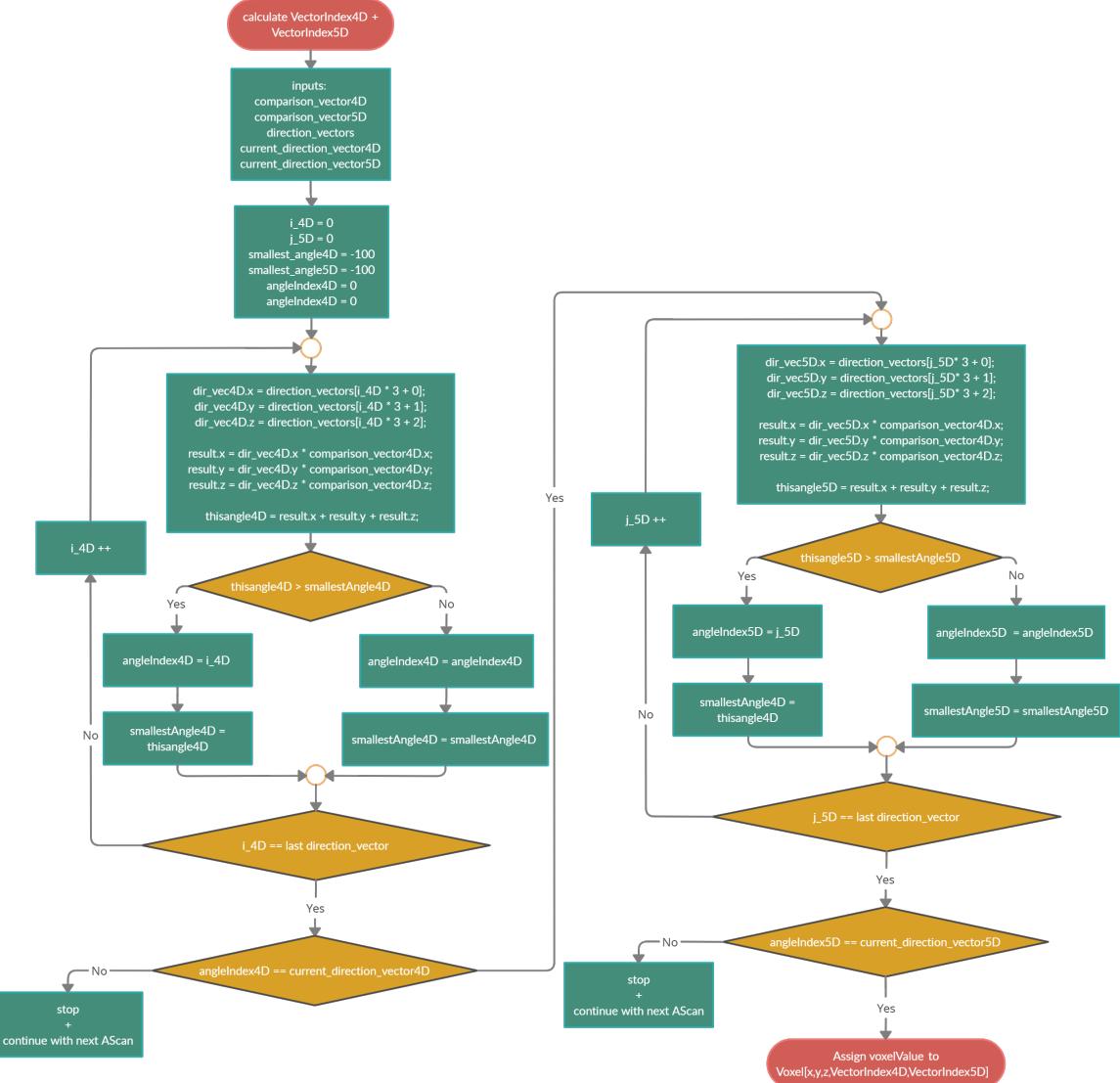


Figure 3.12: Flowchart of the angle sorting algorithm.

The basic approach for the successful assignment of a voxel value to the correct directional vector is finding the directional vector which is closest to the comparison vector of the

particular A-scan and assign the voxel value to this direction. Since there are multiple directional vectors the comparison vector is compared to each one of them. As we are using the orthogonality as metric for the angular relation between two vectors the goal is to find the largest orthogonality between the comparison vector and each directional vector. For the directional vector which yielded the highest orthogonality the voxel value is assigned to this direction.

The input parameters for this approach are the `comparison_vector4D` and `comparison_vector5D` for both dimensions. For this example they will be defined as the vector from the voxel to receiver for the `comparison_vector4D` and the vector from the voxel to the emitter for `comparison_vector5D`. The directional vectors that discretise the directional information of the volume have to be imported as well. They are regarded as input argument for the algorithm as an interleaved array of vector coordinates. The process of calculating the voxel value, the calculation of the comparison vectors and the assignment of these comparison vectors to the directional vectors is repeated two times for every directional vector there is. In each iteration two current directional vectors for the 4th and 5th dimension are given. The indices of the two directional vectors that are under investigation in that particular iteration of the reconstruction are given as `current_direction_vector4D` and `current_direction_vector5D`.

In the 2nd stage of this approach the iterators `i_4D` and `j_5D` are initialised. Furthermore, the variables `smallest_angle4D` and `smallest_angle5D` are initialised as an arbitrary small value. They will later hold the biggest orthogonality (since large orthogonality equals small angle) of the set of directional vectors to the comparison vector. The `angleIndex4D` and `angleIndex5D` are set to zero in the beginning.

In the 3rd stage of this method the directional vectors for the first iteration `i_4D = 0` are calculated. As the coordinates of the directional vectors are stored in an interleaved array, pointer operations can be used to access the coordinates. For '+0' the first coordinate of the directional vector is loaded. Analogously, the second and the third coordinates are taken from the array with '+1' and '+2'. The result is the first directional vector `dir_vec4D` for the first iteration of `i_4D`.

This is the point where the orthogonality comes into play. With the `comparison_vector4D` and the first directional vector `dir_vec4D` the scalar product can be calculated. The result is stored in variable `thisangle4D` which ultimately is nothing else but the orthogonality between the two vectors. The following step is for the decision whether this new `thisangle4D` is bigger than the `smallest_angle4D`. If this is the case the left path is taken and the `angleIndex4D` is set to the current value of the iterator `i_4D` and the new `smallest_angle4D` is set as `thisangle4D`.

Since `smallest_angle4D` was initialised as a very small value the first iteration will always lead to the left path which overwrites `smallest_angle4D`. If the setting of a new `smallest_angle4D` is done the iterator `i_4D` will be incremented by one if the last directional vector was not already reached. Again a directional vector `dir_vec4D` is calculated from the coordinate array and the orthogonality is calculated and stored as `thisangle4D`. Once again the control structure checks if the new `thisangle4D` is

smaller than the `smallest_angle4D` that was set in the iteration before. For the case being bigger again the left path is chosen and the old `angleIndex4D` is replaced by the actual iterator `i_4D` and the old `smallest_angle4D` is replaced by the new `thisangle4D`. In case of `thisangle4D` not being bigger than the previous `smallest_angle4D` at the end of the left side of the flowchart, the right path is taken. This leaves the `angleIndex4D` and the `smallest_angle4D` unchanged. This procedure is repeated for every directional vector there is. Upon reaching the last iteration in the variable `smallest_angle4D` there is saved the orthogonality between the `comparison_vector4D` and its closest directional vector. The directional vectors index is stored in `angleIndex4D`.

After comparing the the `angleIndex4D` with the `current_direction_vector4D` there are two options.

The first is for the case that the `angleIndex4D` and the current `current_direction_vector4D` do not match. It was mentioned that the `current_direction_vector4D`, which is given as input for the algorithm, contains the index of the directional vector that is analysed for that iteration of the algorithm. This means that only feasible values for that particular `current_direction_vector4D` are accepted and every other direction is disregarded for that iteration. For that case of `angleIndex4D` not matching the `current_direction_vector4D` the algorithm will be stopped and the next A-scan will be analysed. This leads to a new `comparison_vector4D` and possibly to a new `angleIndex4D`.

The other option is that the `angleIndex4D` and the `current_direction_vector4D` actually are the same. This means that at least for the fourth dimension the `angleIndex4D` is the directional vector that is regarded for that iteration. In this case the flowchart on the right side of Figure 3.12 will be followed. It shows exactly same procedure but this time for the 5th dimension. In the end there also is an `angleIndex5D` which will be compared to the `current_direction_vector5D`. If they coincide the directional vector `dir_vec4D` will be assigned to `angleIndex4D` and analogously the directional vector `dir_vec5D` will be assigned to `angleIndex5D`. From there the algorithm in Figure 3.9 continues with the next A-scan.

3.5.2.1 Decision area for the angle sorting approach

For the angle sorting approach the decision area for each directional vector is given by the geometrical boundaries of the geometrical arrangement of the directional vectors. For the cases of the platonic solids these geometries are known. An example for the dodecahedron is given in Figure 3.13. In that example six pentagonal faces of the dodecahedron are shown. The directional vectors are plotted in the middle of each face. The assignment of an comparison vector to a directional vector only depends on the face that would be pierced by the comparison vector. The particular face then leads to the directional vector and its index is assigned to that comparison vector. The whole area of each pentagon serves as decision area and the decision boundaries are the edges of the pentagon. Consequently, comparison vectors that pierce the face at a 'tip' where two boundaries meet have a larger angle to the

actual directional vector than a comparison vector that pierces at the middle of a boundary of the same face. Still, both comparison vectors would be assigned the same index regardless of their different angles to the directional vector.

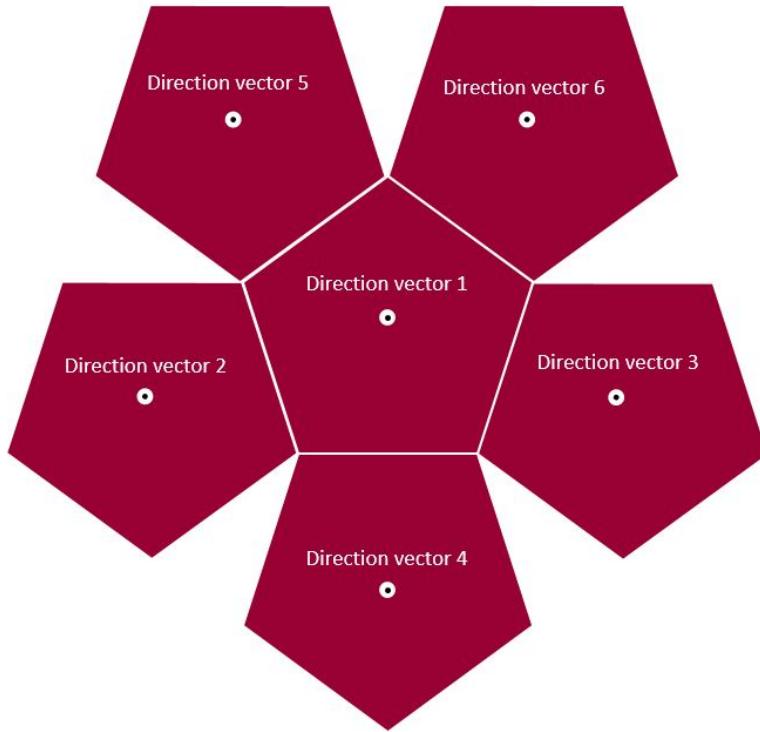


Figure 3.13: 2D example of the decision areas for the partial dodecahedron. Depending on what face of the geometry is pierced by the comparison vector it gets assigned the index of the corresponding directional vector of that face. The decision boundaries for each directional vector are the edges of the particular pentagon.

The decision boundaries for $N \neq \{6, 12, 20\}$ particles do not lead to any known platonic solid geometries. For those unknown cases the angle sorting algorithm also assigns the comparison vector to a particular directional vectors index but the decision boundaries are not as easy to plot in those cases.

3.5.3 Orthogonality threshold method

A new method for the assignment of directional information is introduced in this chapter. Instead of comparing each individual directional vector with the comparison vector in order to search for the smallest angle between them, the new method calculates a threshold value beforehand and compares the orthogonality of the comparison vector to this threshold. A big advantage of this new approach is that it can be implemented in parallel where multiple

execution units could divide the workload by only regarding a certain set of direction vectors. The previous implementation needed to be executed sequentially since every directional vector had to be regarded for the angle sorting step.

3.5.3.1 Decision area for orthogonality threshold

In the case of the generation of an arbitrary number of directional vectors generally there are no known geometries. In theory the angle sorting algorithm would also lead to a decision on each directional vector. Since this criterion is hard to verify for unknown geometries the goal is to find a new criterion for all directional vectors which lead to the same boundaries. These areas of decision are chosen to be cone shaped with the opening angle of the cones relating to the angle of the bisector between two normals. An example for that is shown in Figure 3.14:

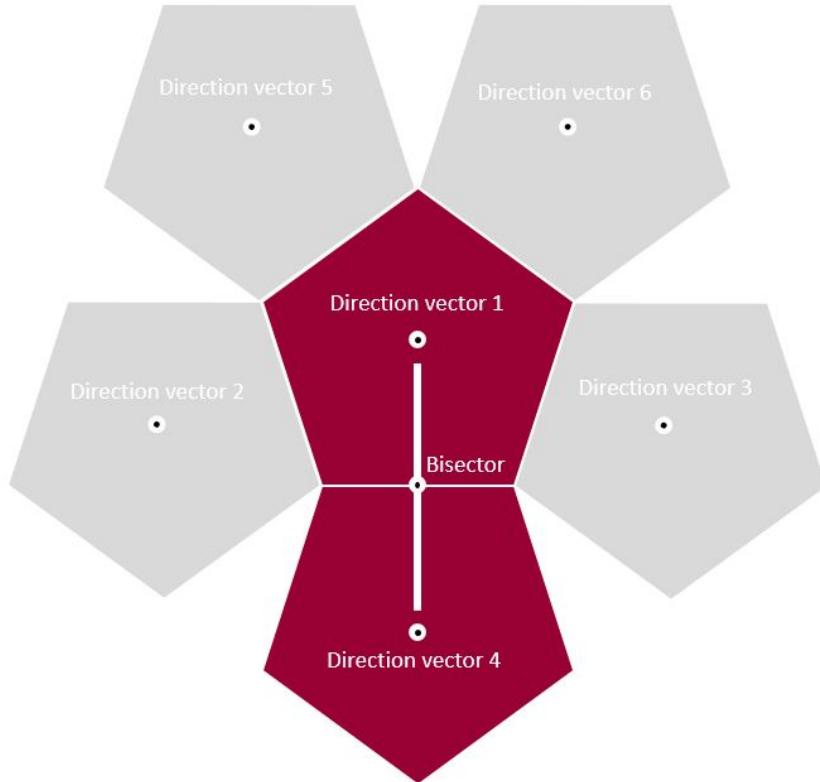


Figure 3.14: Calculation of the bisector between the nearest neighbour of the first normal.

The decision areas for this method are shown in context of the decision areas of the dodecahedron. The first step to find an orthogonality threshold for the decision criterion is to find the nearest neighbour of the first directional vector by the means of orthogonality. This

is shown in Figure 3.14. The orthogonality is calculated for each combination of the first directional vector with each other directional vector in the geometry. For the pair with the highest orthogonality it can be assumed that this is the approximately nearest neighbour by means of the angle between them. Since this nearest neighbour is only searched for the first directional vector this results in an locally optimal solution. Theoretically, every combination with each directional vector has to be tested to find the global minimum angle. Of course by not regarding every combination of directional vectors it can not be guaranteed that the chosen radius of the decision areas prevents a certain overlap of the regions. Since the directional vectors are equally distributed, the local solution for the nearest neighbour yields an acceptable solution. In this example the closest pair of vectors is the first and the fourth one. Between those two vectors the normalised angle bisector is constructed. After this step the orthogonality between the first directional vector and the angle bisector is calculated. This value is the orthogonality threshold for the algorithm. As was shown in section 3.5.1 the absolute angle between two vectors is not important when it comes to assigning a directional index to it. If the orthogonality between a comparison vector and the directional vector is above the orthogonality threshold this comparison vector will be assigned to that directional vector. In the 2D example the decision boundaries can be plotted as circular areas around the directional vectors. This is shown in Figure 3.15:

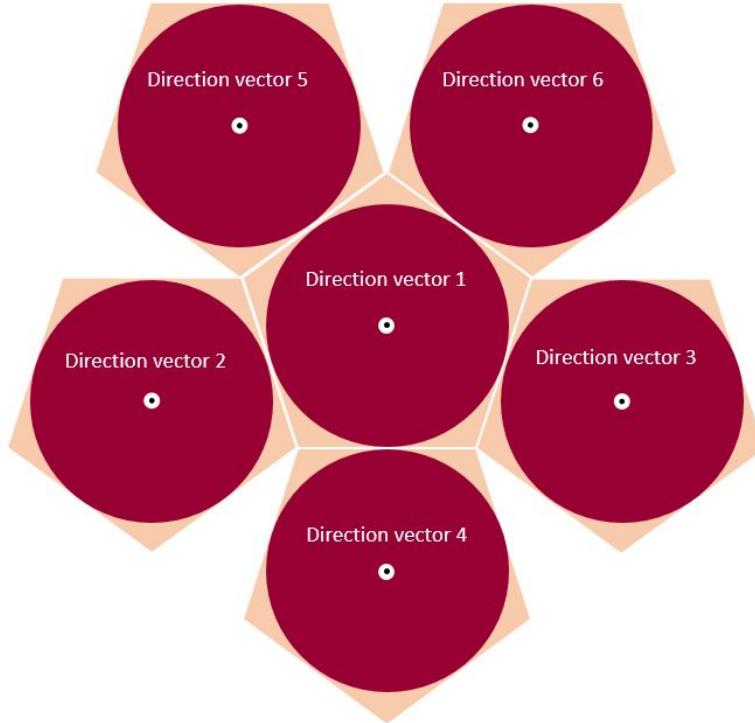


Figure 3.15: Circular decision areas around each directional vector.

The red circles mark the decision areas where a potential comparison vector would be assigned to the directional vector in the middle of that particular circle. Between those circles the previous decision areas for the angle sorting method are visible. It becomes clear that some areas of the pentagon are not part of the decision area anymore. Every comparison vector that misses the circular decision areas is not assigned to a directional vector at all and will not be part of the reconstructed image. It was mentioned above that this is not an inherent problem of the method. If the decision areas would be overlapping no A-scan would be discarded.

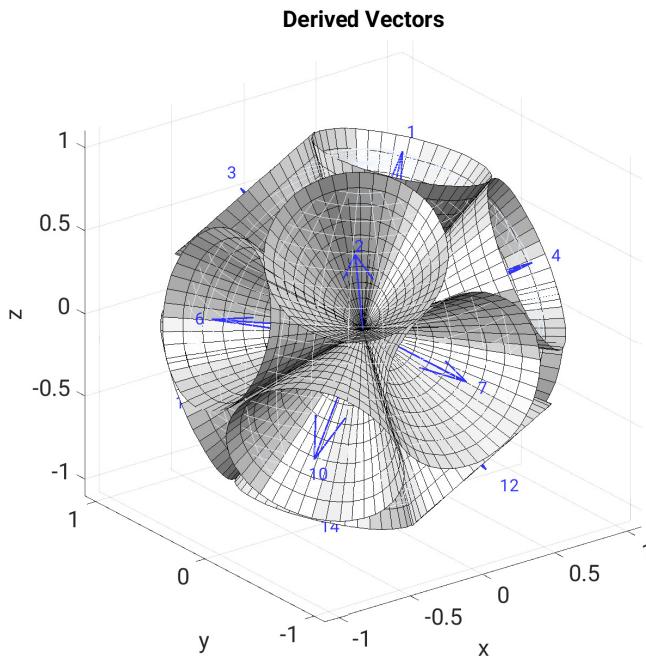


Figure 3.16: Decision areas for the orthogonality threshold in 3D depicted as cones.

Figure 3.16 shows an extension of the directional vectors which were shown in in Figure 3.8 on the right. For each directional vector the decision cone was plotted and it shows the boundaries of each directional vector. Each cone has the opening angle which corresponds to the angle between the first normal and the bisector between the first directional vector and its nearest neighbour. Each comparison vector which lays inside one cone would be assigned to the index of the corresponding directional vector in the middle of that cone.

3.5.3.2 Assignment of directional dimensions with orthogonality threshold

The new technique for assigning an directional index to a certain comparison vector is introduced in this section. It has some similarities to the previous angle sorting approach but differs from it in context of performance. The principle is explained with the following flowchart:

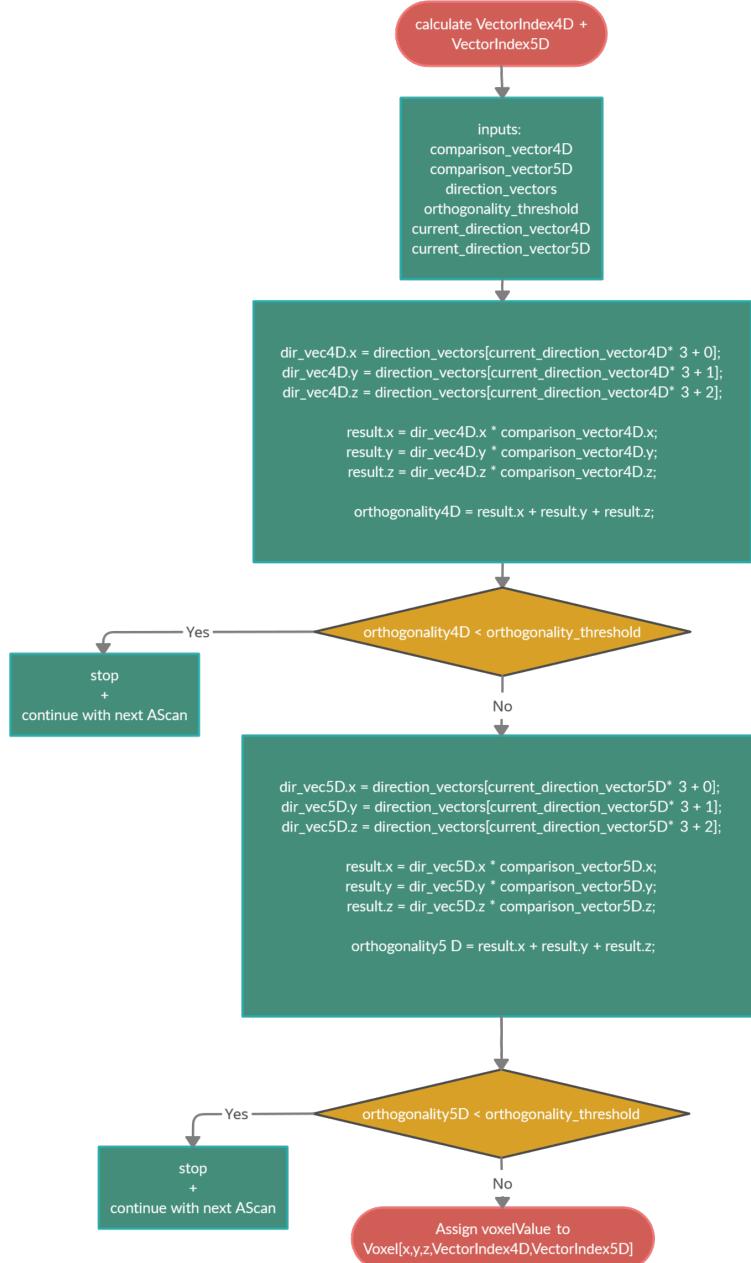


Figure 3.17: Flowchart of the orthogonality threshold algorithm in 5D.

The input parameters for this approach are the two **comparison vectors** for both dimensions. Since the definition of the comparison vectors is free, for this example they will be defined as following: the `comparison_vector4D` is defined as the vector from the voxel to receiver. The other `comparison_vector5D` is defined as the vector from the voxel to the emitter. The set of **directional vectors** that discretise the directional information of the volume are also part of the input values in form of an interleaved array of coordinates. Next the threshold value as decision criteria for each directional vector is imported. The calculation of the **orthogonality threshold** is explained in section 3.5.3.1. The indices of the two directional vectors that are under investigation in that particular iteration of the reconstruction are given as `current_direction_vector4D` and `current_direction_vector5D`. In the 2nd stage of the procedure the coordinates of the directional vectors for the 4D case have to be chosen from the set of directional vectors, since `current_direction_vector4D` and `current_direction_vector5D` were only given as indices. As the coordinates of the directional vectors are interleaved in the array they can be accessed with the pointer operation that is shown in the 2nd step of the flowchart. For '+0' the first coordinate of the directional vector is loaded. Analogously, for '+1' and '+2' the second and the third coordinates are taken from the array. With that we have the current directional vector `dir_vec4D` for that iteration. The scalar product between the `comparison_vector4D` and the directional vector `dir_vec4D` is calculated in the last part of the 2nd stage. The resulting `orthogonality4D` is compared to the `orthogonality_threshold` which was part of the input parameters. If the `orthogonality4D` is smaller than the `orthogonality_threshold` it can be assumed that the `comparison_vector4D` does not belong to the currently investigated `dir_vec4D`. In this case the algorithm will stop and continue with the next A-scan where a new `comparison_vector4D` is calculated. If the condition leads to the conclusion that the `orthogonality4D` is larger than the `orthogonality_threshold` the next stage of the algorithm is started. The procedure is repeated for the 5th dimension in the same manner leading to the `orthogonality5D`. If `orthogonality5D` actually is larger than the `orthogonality_threshold` the voxel value will be assigned to the voxel position[x,y,z] and to the corresponding 4th and 5th dimension of the output volume. If not, the algorithm again will stop before any voxel value can be assigned to any part of the image and the next A-scan will be analysed.

3.6 Performance implications of assignment approaches

In theory the voxel volume can be segmented into an arbitrary set of directions with a very high density of directional vectors. To find the corresponding vector index for each A-scan the process of index identification which is explained in section 3.5 has to be repeated for every emitter-receiver combination, for each voxel and each aperture position. Furthermore, the calculation has to be repeated for each directional vector twice. Depending on which geometry or what number of N particles is chosen for the generation of directional vectors,

this could lead to a large amount of data that has to be processed. The average reconstruction of a three dimensional measurement volume results in

$$\#Calculations = \#Voxel \cdot \#Emitter \cdot \#Receiver \cdot \#AperturRotation \cdot \#Vectors^2$$

number of calculations. For the case of using a 12 faced dodecahedron, 628 emitters, 1413 receivers, a volume of 250x250x250 voxels and ten aperture positions $[12^2 \cdot 628 \cdot 1413 \cdot 250^3 \cdot 10] = 1.982 \times 10^{16}$ calculations have to be performed to find the smallest vector index. Therefore, the performance impact of the algorithms for the identification of the index is non-negligible. By decreasing the complexity of these calculations the performance of the reconstruction algorithm can be greatly improved. The goal is to receive a qualitative performance for each iteration of the two assignment methods and to compare them on the basis of these findings. The results of the performance analysis are presented in section 4.4.

Results

4.1 Experimental Setup

For the evaluation of the algorithm a suitable data set of a setup with a simple object with a smooth, organic surface was chosen. The following data set 'exp0040_Carina-PhantomGelatineOlive_mitHalterung' fulfils these requirements [26]. The acquired data resulted from an USCT measurement of a phantom consisting of an olive inside a gelatin block. For the imaging a phantom was placed into the USCT device. The measurement setup can be seen in Figure 4.1:

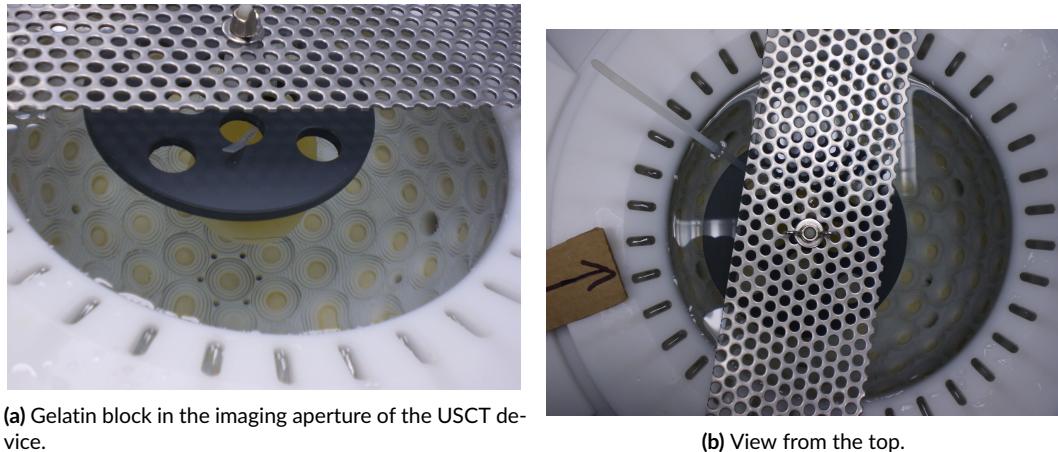


Figure 4.1: Measurement setup of the gelatin block with olive inside of the USCT device.

The A-scans were recorded for all 157 TAS with ten aperture positions so a total of $157 \cdot 4 \cdot 157 \cdot 9 \cdot 10 = 8.8 \cdot 10^6$ A-scans were recorded. The reconstructed reflection image for this data set is shown in Figure 4.2. The sub plots each show a slice through the image. The coloured lines in each plot show the selected slice in the other two coordinate systems. The blue marker belongs to the z-dimension, the green one corresponds to the y-dimension and

the red marker shows the x-dimension. Figure 4.2a shows the x-layer for $x \approx 0$. That x is equal to zero can be seen in the other two figures where the red markers are located at approximately 0. Analogously, Figure 4.2b and 4.2c show the y-layer and z-layer of the image.

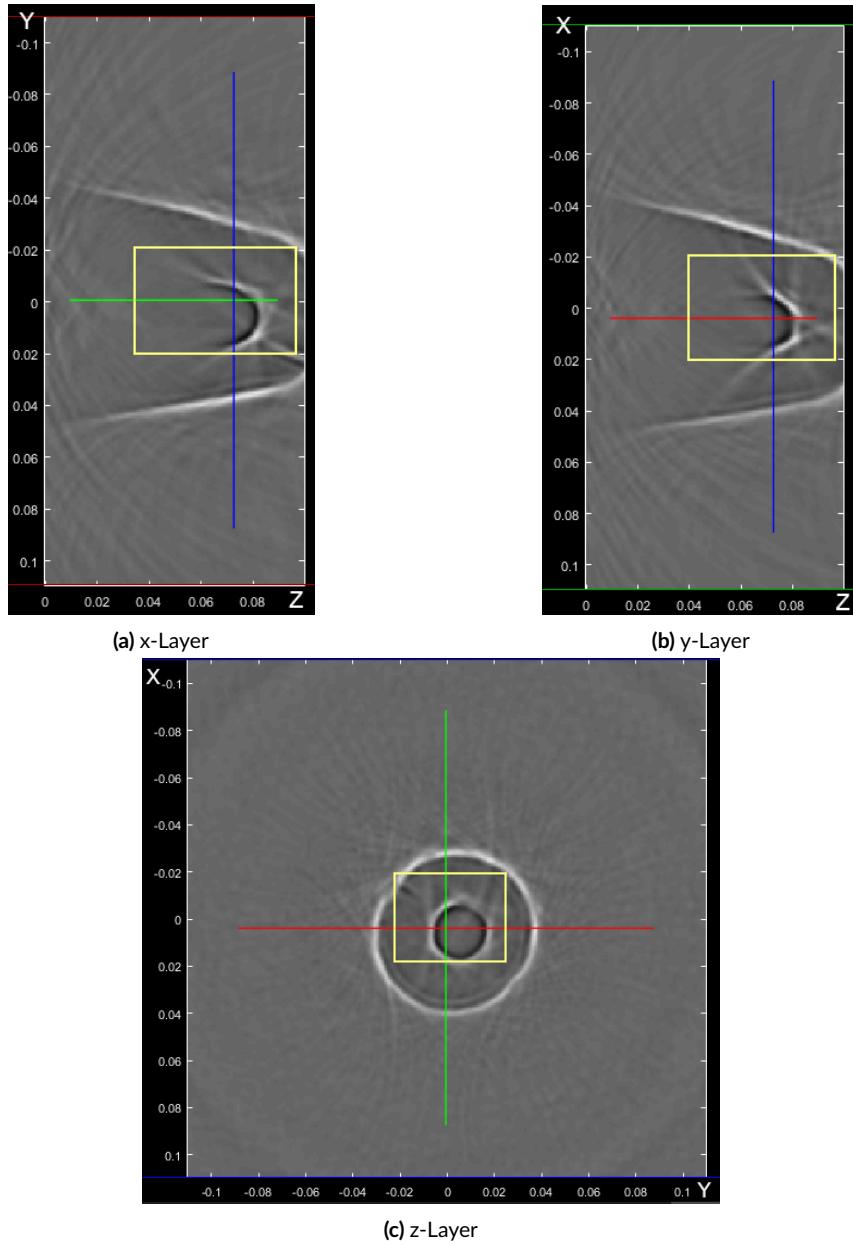


Figure 4.2: Reconstructed reflection image of an olive in gelatin. The yellow box marks the area that was used during the reconstructions with five dimension to limit the amount of data.

In the centre of Figure 4.2c the olive can be seen as the smaller, darker circular shape. The brighter shape around the olive is the gelatin block.

The volume was only partially considered during the 5D reconstruction to reduce the amount of data that had to be processed and by that also reduce the computation time. The start point for the volume was set to $[-0.02, -0.02, 0.04]$ and the end point was set to: $[0.02, 0.02, 0.095]$. The volume considered in the analysis is marked with the yellow boxes in Figure 4.2. This particular volume was chosen to include as many different types of tissue as possible. Therefore, four particular voxels where chosen at the location of different tissue samples. The first voxel is located at $[85, 66, 87]$ at the edge of the olive. The 2nd test voxel is located at $[18, 21, 87]$ right in the gelatin around the olive. The third voxel is located on the surface of the gelatin block at the coordinates $[1, 6, 76]$. At $[40, 60, 87]$ another voxel was chosen in the pulp of the olive. The position of these voxels can be seen in Figure 4.3.

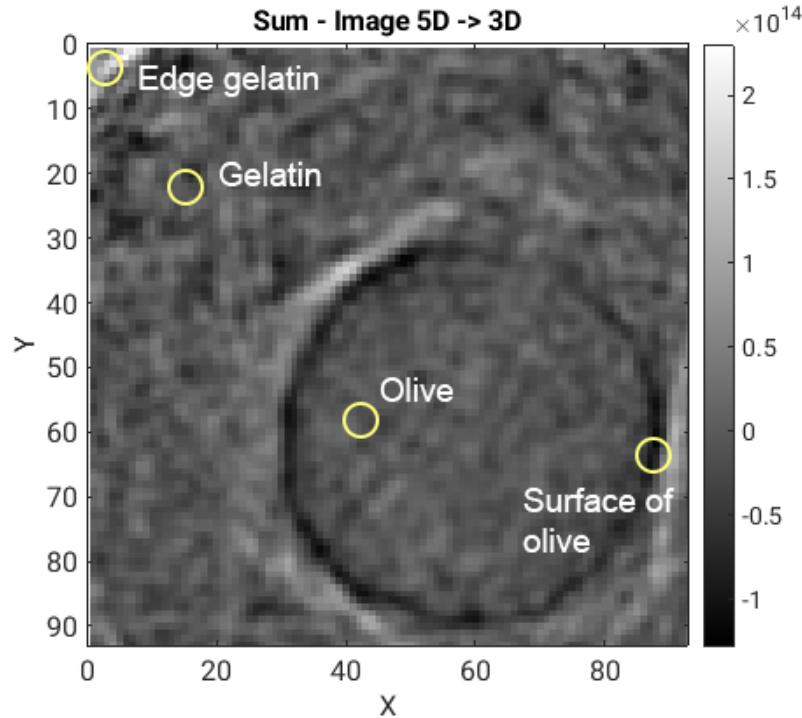


Figure 4.3: Overview of the reconstructed area with the location of the test voxels for the different materials inside of the imaging aperture.

4.2 Evaluation of the assignment process during 5D reconstruction

The introduction of the fifth dimension was explained in section 3.1. The assignment process of the voxel value V_k to the voxels in the 4th and 5th dimension was explained in section 3.5. The functionality of this process is verified in this section. In this example the 4th dimension will relate to the comparison vector from the voxel to the receiver. The 5th dimension is defined as the comparison vector from the voxel to the emitter. During the reconstruction 14 directional vectors were used. The configuration of active emitters and receivers is shown in Figure 4.4. The green crosses mark the location of all active receivers and the red circles represent the emitters. For this example the first 30235 A-scans are used which include the first 40 emitters and 1309 receivers. The data of every receiver that lays in a $\pm 120^\circ$ angle to the sender normal is included for that particular emitter.

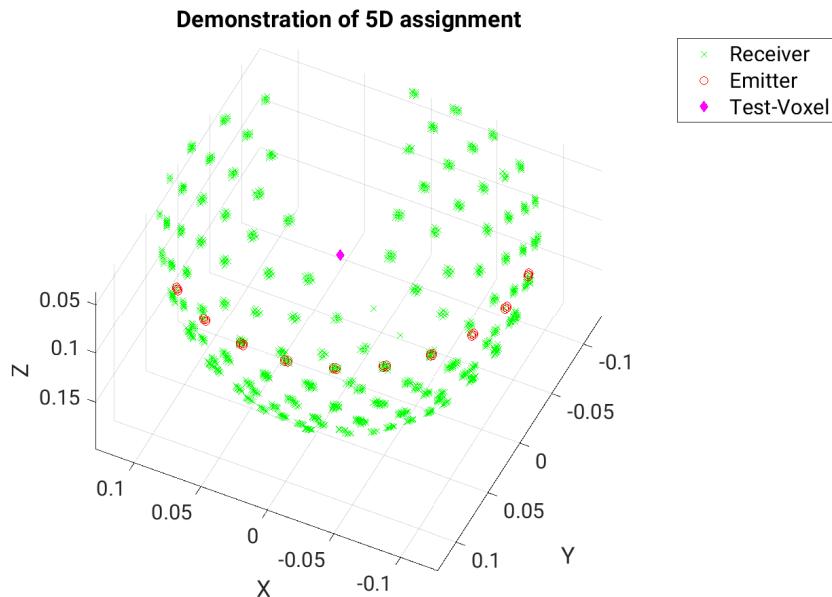


Figure 4.4: Configuration of 10 emitting and 145 receiving TAS and a test voxel in the centre of the aperture. The receivers are shown in green and the emitters as red circles. The units on the axes are given in meters.

For this example a test voxel is defined arbitrarily in the centre of the aperture. The coordinates of this point are given by voxel indices [150, 150, 150] and it is shown as the pink diamond in Figure 4.4. The coordinates of the voxel can be converted into the coordinate system of the USCT which is given in meters. The test voxel therefore is located at [0.0047m, 0.0047m, 0.0047m].

The following figure shows the voxel values for the one test voxel in the centre of the aperture at the position of the pink diamond. Since there are 5 dimensions each dimensions gets assigned a certain amount of voxel values for each combination of dimensions. Every combination of the 4th dimension with the 5th dimension is shown in Figure 4.5.

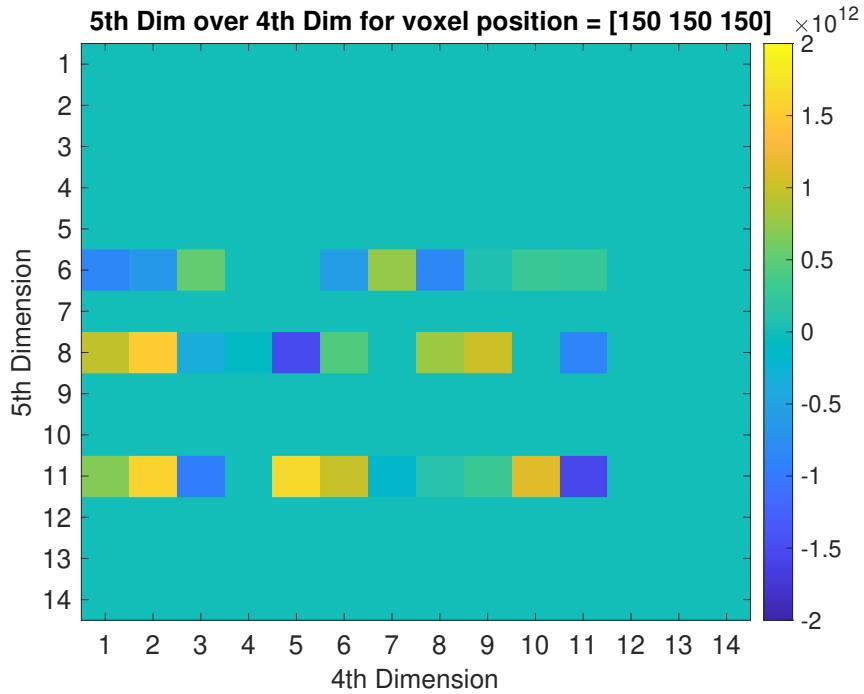


Figure 4.5: Resulting voxel values V_k of the five dimensional reconstruction. The 4th dimension relates to the voxel-receiver-vector. The 5th dimension represents the directional information for the voxel-emitter-vector.

To understand Figure 4.5 the example of the Rubik's cubes in Figure 3.4 shall be used. In the previous example there were only four directional vectors. This resulted in $4 \times 4 = 16$ Rubik's cubes. In the 4th dimension the cubes held the information about the receiver directions. The the 5th dimension held the data for the emitters. The here shown data was generated with 14 directional vectors which leads to a 14×14 matrix with a total of 196 Rubik's cubes. These 196 Rubik's cubes are shown in Figure 4.5 but not as whole but only the value of the voxel $[150, 150, 150]$ in each cube. Every value that can be found in the same row of the matrix in Figure 4.5 was recorded by the same emitter. Analogously, every value in the same column belongs to the same receiver. The distribution of values in the 5D-over-4D representation is compared to the emitter-receiver-configuration from Figure 4.6. The 14 directional vectors and their corresponding decision cones are shown.

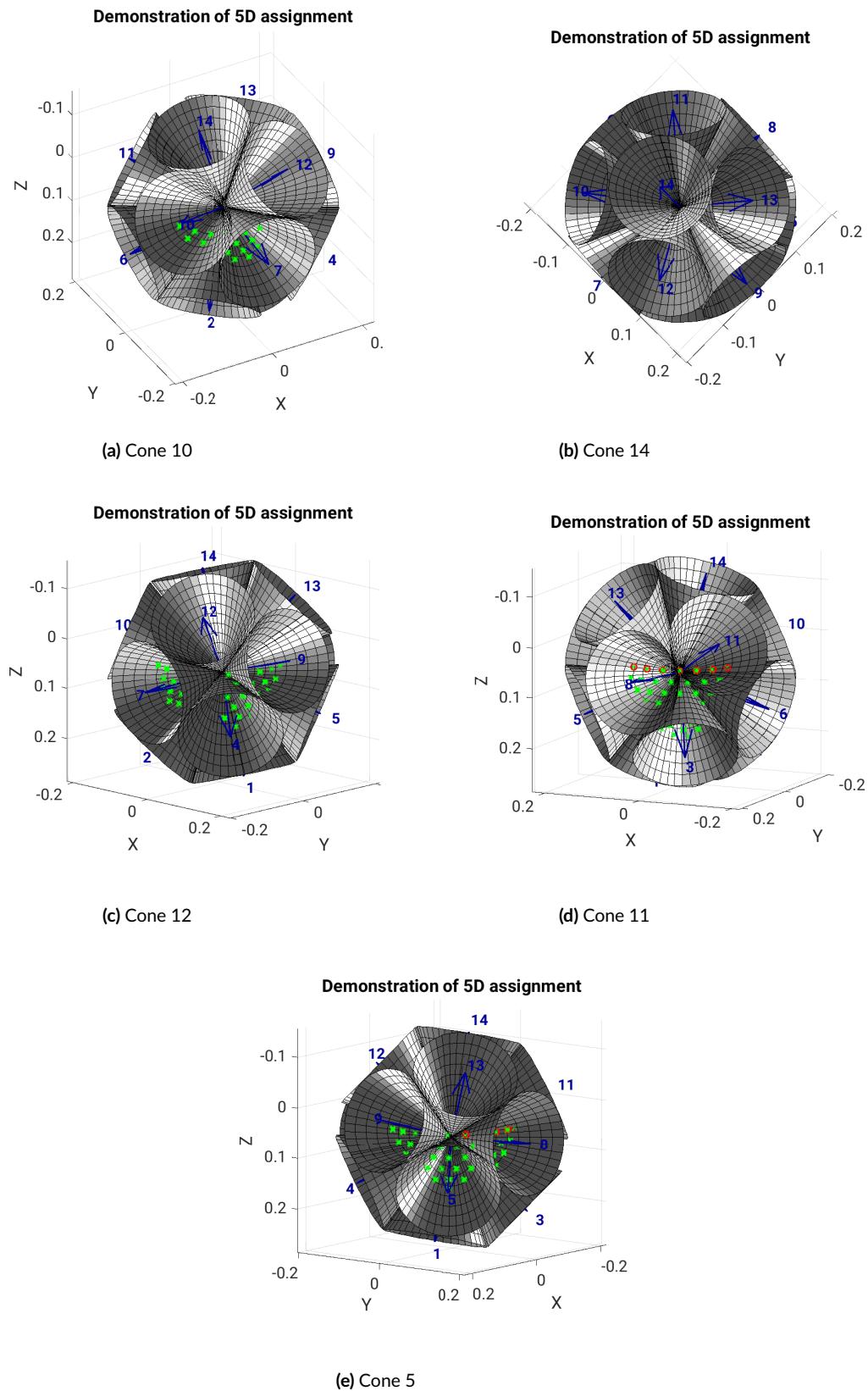


Figure 4.6: The emitter receiver configuration of Figure 4.4 with added directional vectors and decision cones. The origin of each cone is located at the position of the test voxel [150, 150, 150].

To connect the results shown in Figure 4.5 with the cone images in Figure 4.6 we will begin with the interpretation of the the emitter information. The emitters that were used are represented by the red markers in Figure 4.6.

The cones show the decision areas of each directional vector. If a emitter position can be seen inside of a cone (i.e. a red dot is visible in the cone) the voxel values V_k originating from that emitter are assigned to the directional vector of that cone.

In this example the following cones contain emitter positions: 6, 8 and 11. The results in Figure 4.5 show that only the rows for corresponding to emitters 6, 8 and 11 are occupied. All other rows are empty. This coincides with the assumption that the 5th dimension relates to the voxel-emitter directional information the voxel values are only assigned to the corresponding emitter cones.

For the receivers, which are plotted in green, the data is stored in the 4th dimension of Figure 4.5. This coincides with the representation in Figure 4.5 in which all columns are occupied, for which receivers were assigned to the directional cones.

There are cones that point towards the opening of the aperture where no transducers are located. Figure 4.6b shows an example of cones which contain neither emitters nor receivers. The empty cones in this example are 12, 13 and 14. Therefore, no data is recorded for these cones. In the 5D-over-4D representation in Figure 4.5 the voxel values of these empty cones are empty. For the indices 12, 13 and 14 neither the columns nor the rows are occupied.

To verify that the image information was assigned correctly during the reconstruction, the following figure shows the sum image which was generated by summarising all values over the 4th and 5th dimension. By that we should receive an image, which is equivalent to a standard SAFT image neglecting the directional information.

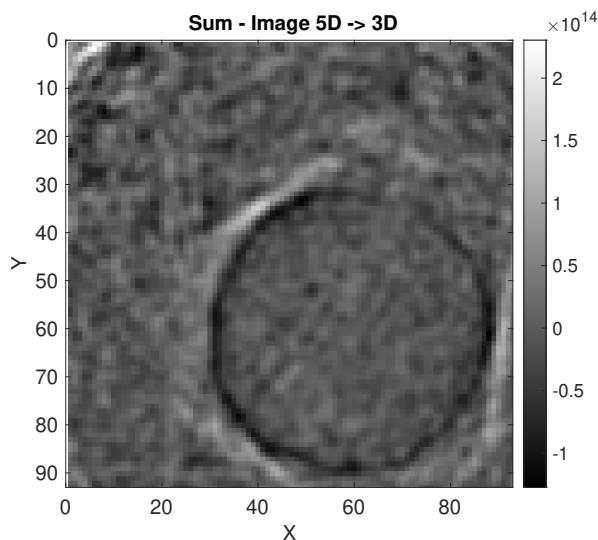


Figure 4.7: Sum image of the five dimensional data over the 4th and 5th dimension.

The sum image shows the 3D image of the olive and therefore the expected outcome was reached (see for comparison Figure 4.2). If something had gone wrong during the assignment of the directional information, strong visible artefacts would be expected in the sum image that would exceed the expected artefacts of the SAFT image.

4.3 Comparison of orthogonality threshold method and angle sorting method

In section 3.5 two methods for the assignment of comparison vectors to the right directional vector were explained. It was also mentioned that both methods have different decision regions for the directional vectors. The influence of these differences is shown in this section. In the following all images were created using 25 directional vectors and with the added 4th dimension which stores the information about the voxel-receiver relation. The reason for not also including the 5th dimension was to keep the computation time as small as possible and still being able indicate the differences in the resulting images. Figure 4.8 shows a side by side comparison of the reflection image for both assignment methods. From the 4D image the following images are from the 3rd directional vector and the 166th slice in the z-dimension.

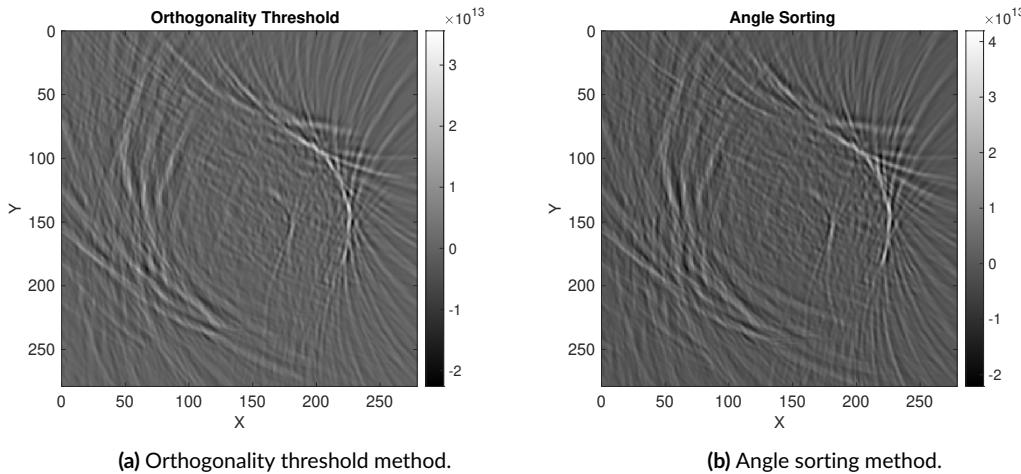


Figure 4.8: Side by side comparison of the resulting images for both methods. 25 directional vectors in total were used. The images are shown for the 3rd directional vector and the 166th slice in the z-dimension.

Figure 4.8b on the right shows a slightly higher contrast than its counterpart on the left. This can be seen in the legend on the right. The highest voxel value for the orthogonality threshold method results in approximately 3.7×10^{13} where as the amplitude for the angle sorting method is a bit higher at 4.1×10^{13} . The reason for that is that the angle sorting method of section 3.5.2 assigns every available A-scan to a directional vector whereas the orthogonality

threshold method discards every A-scan for which the comparison vector lays between the decision cones.

It has to be remarked that it is not an inherent characteristic of the orthogonality threshold method to discard a certain amount of A-scans. For this thesis it was implemented in a particular way so that the decision regions of all directional vectors are all the same size and overlap as little as possible. This implementation leads to as little ambiguity as possible concerning the directional assignment. The non-overlapping decisions regions that are schematically shown in Figure 3.15 lead to a consistent quality of the assignment with the same decision criteria for each directional vector.

For the angle sorting method the decision criteria resulted in different acceptance angles depending on whether the comparison vector points towards a 'tip' of two boundaries of the pentagon or if they were located towards the bisector between two directional vectors. In these cases the quality of the assignment can vary.

In the following image the two methods were compared for one voxel in each of the 25 volumes in the 4th dimension. This representation is generated in a similar manner as the 5D-over-4D representation in Figure 4.5 was generated. Since there is no 5th dimension this time only the values for the 4th dimension are plotted. For this representation the analogy of the 4D Rubik's cube array in Figure 3.2 can be taken. Instead of the four Rubik's Cubes we now have 25 and for each of those cubes the voxel value at the coordinates [150, 150, 166] is displayed. For the directional vectors 22 to 25 no or only very few values are shown. The reason for that again is that those directional vectors are pointing out of the aperture and there are no receivers that could detect or emit a signal.

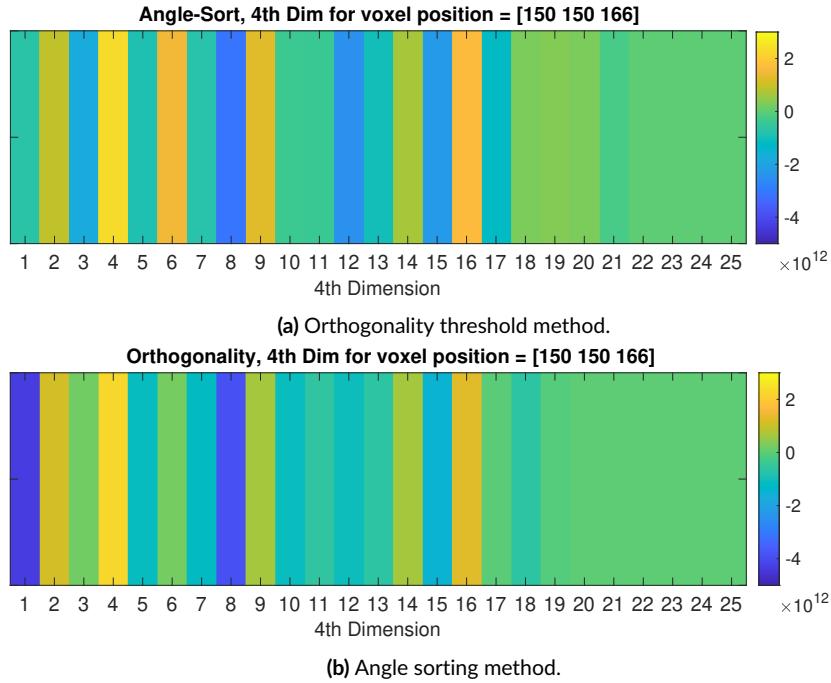


Figure 4.9: Comparison of each voxel value for each of the 25 directional vectors.

The overall structure of the array of voxel values for both methods visually appear to be similar. Still, there are some differences e.g. for the 1st directional vector. The voxel value for the orthogonality threshold method was -4.3171×10^{12} where as the voxel value for the angle sorting method was -7.1738×10^{11} . Hence, approximately six times higher than the angle sorting value. For a better comparison of the results shown in figure 4.9, in figure 4.10 each voxel value for the total of 25 directional vectors is shown in a single plot. The red line shows the voxel values which were yielded by the orthogonality threshold approach. The blue graph belongs to the angle sorting method. In most points the blue line lies above the red line. This corresponds to the assumption that the orthogonality threshold method does not consider every A-scan and therefore this method yields images with an overall lower voxel values.

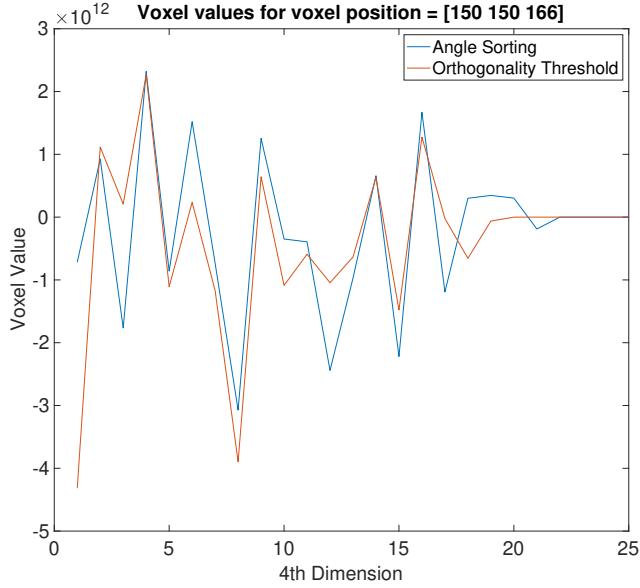


Figure 4.10: Each voxel value of Figure 4.9 as diagram. The blue graph belongs to the angle sorting Method and the red graph to the orthogonality threshold.

To make an assessment about how big the influence of the reduced contrast of the orthogonality method is, the reconstructed 4D images are both reduced to a 3D image. For each of the 25 sub-volumes the same voxel position is considered. This leads to 25 voxel values for one particular voxel in each volume. The mean of those 25 voxel values is calculated and written back into the coordinates of the particular voxel. This is repeated for every voxel there is hence the final images shows approximately the outcome of the standard SAFT without considering the directional information. The final images are shown in figure 4.11.

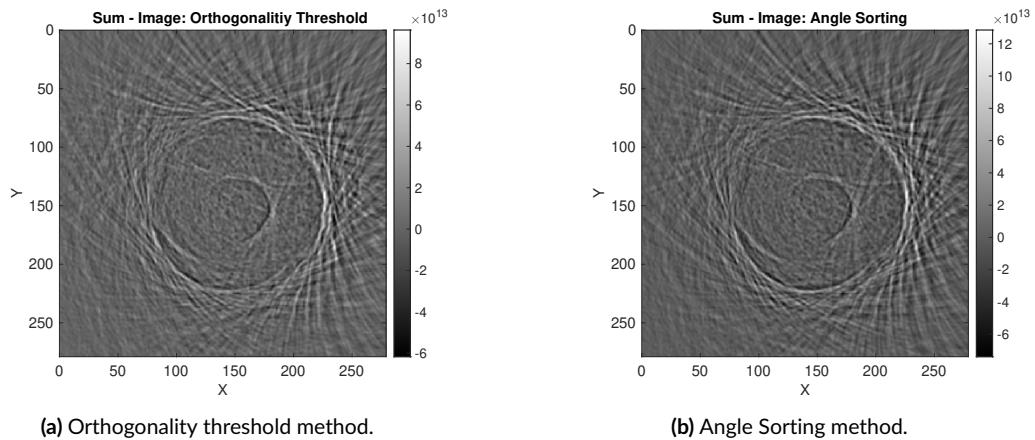


Figure 4.11: Summarised image where the 4D image was reduced to a 3D image.

Again the 166th slice in the z-direction is shown. In comparison to Figure 4.8 the olive in the middle of the volume becomes much more prominent since reflections from all directions are considered. In the image of only one directional vector the olive in Figure 4.8b showed a high intensity of sound waves in the top right corner of the olive. The outline of the gelatin block around the olive becomes more prominent as well. With the summation of all the images the directional information is lost however. Both images in figure 4.11 contain enough detail to make out the gelatin block with the olive in the middle. A difference concerning the noise level can not be observed. Again the angle sorting method results in a higher amplitude compared to the orthogonality threshold. The orthogonality threshold has a maximum voxel value of approximately 9×10^{13} whereas the amplitude of the solution of the angle sorting method in Figure 4.11b reaches 8.7×10^{13} .

A qualitative representation of the differences between both sum images is given in Figure 4.12.

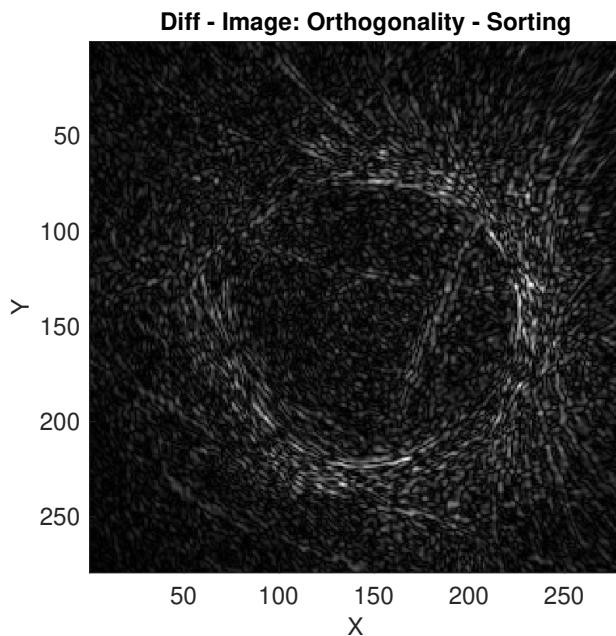


Figure 4.12: Absolute value of the difference between the summarised image of the Orthogonality Method and the Angle Sorting method.

The difference image shows that the inside of the gelatin block and the olive is mostly unaffected by the new approach. Big differences can only be observed on the outer boundary of the gelatin block. The olive can not be distinguished in the difference image. Therefore, the new methods does not influence that part of the image which is a desirable result. The new approach leads to a different amplitude in the reconstructed images compared to angle sorting approach. However, the part of the image that contains valuable information is not influenced at all.

4.4 Performance of the directional dimension assignment methods

In Section 3.5 two main approaches were presented for the assignment of the directional vector index to each A-scan. In this chapter the results of the performance evaluation of the angle sorting approach from section 3.5.2 and the threshold orthogonality from section 3.5.3 are shown.

For the performance analysis the general structure of the reconstruction from Figure 3.9 is adapted. To speed up the process, the 5th dimension is kept constant during the execution. This leads to a four dimensional image instead of a five dimensional. For the evaluation a script was written which calls the reconstruction function with an increasing amount of directional vectors. For each choice of the number of directional vectors the four dimensional image is reconstructed in whole. The procedure is repeated for the angle sorting approach and the new orthogonality approach separately.

The following calculations were performed on two GPUs (Nvidia GeForce 2080 Ti) with a reconstruction volume of $279 \times 279 \times 233$ voxels. Furthermore, a block of 30235 A-scans were used during the reconstruction to speed up the process. The four dimensional case was considered and therefore only the comparison vector from the voxel to the receiver was used. No results besides the profiling report were saved to minimise the influence of hard drive operations during the performance evaluation. The execution time includes all initialisation steps of the reconstruction algorithm as well as generation of directional vectors. The calculation of the threshold orthogonality (in case of the application of the orthogonality threshold method) is also considered by these results. The first reconstruction was started for 14 directional vectors and the last ended at 126. The current implementation allows to generate an arbitrary set of directional vectors starting at 14. Sets with fewer directional vectors than 14 were generated with the platonic solid approach. For the performance evaluation it was necessary to increase the number of directional vectors arbitrarily and the evaluation starts with 14 input vectors.

The following three figures shows the comparison of the computation time of the different techniques for the index identification introduced in section 3.5. The results for the sorting algorithm are presented in blue where as the approach with the orthogonality threshold is shown in red.

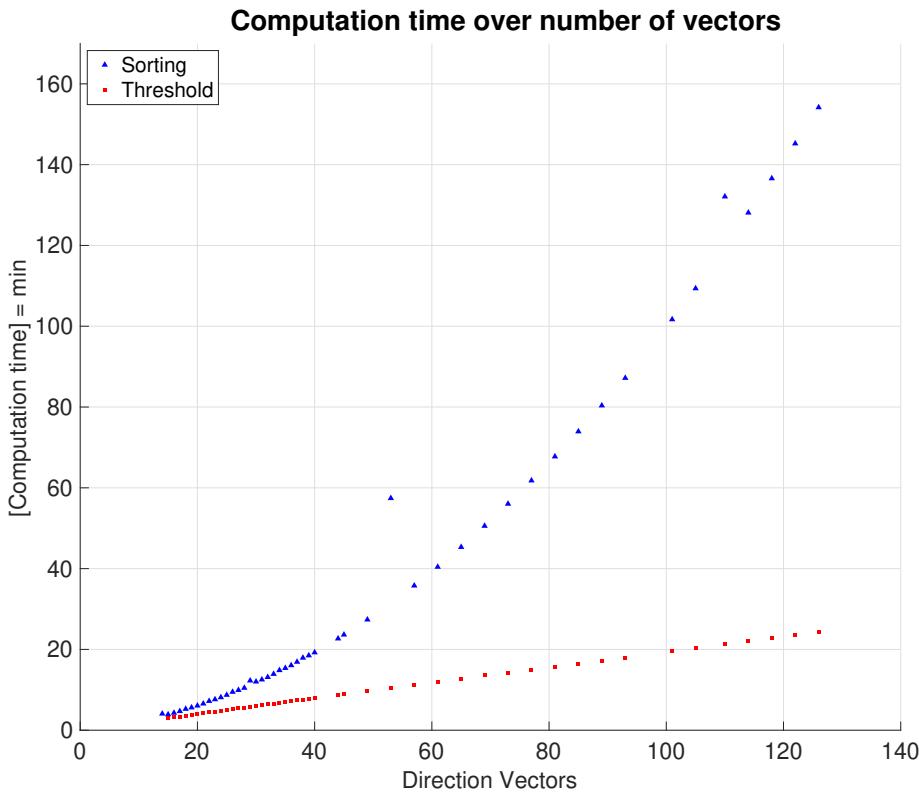


Figure 4.13: Total time of computation given in minutes for 30325 A-scans and for all loops over the directional vectors. The red markers represent the execution time for the orthogonality threshold method. The blue markers belong to the angle sorting algorithm.

The blue triangle markers in Figure 4.13 represent the execution time for the angle sorting algorithm. The horizontal distances between the succeeding markers is associated with the attempt to decrease the overall length of the experiment. For shorter execution times (up to 20 minutes per call) for both algorithms the amount of directional vectors were increased one by one after each iteration. The following steps were increased by four input vectors per iteration.

For a small number of directional vectors the execution time of the angle sorting algorithm and the orthogonality threshold are relatively close, both taking approximately three minutes to compute. With increasing number of directional vectors the application of the angle sorting algorithm leads to a longer sorting loop. This can be seen in the exponential growth of the execution time for the implementation of the angle sorting method. In contrast the implementation of the orthogonality threshold method shows a linear behaviour when the number of directional vectors is increased. For the final 126 directional vectors the sorting algorithm takes 154 min whereas the orthogonality method takes 24.3 min to finish, thus being 10.5 times higher than the faster orthogonality method. For 45 directional vectors the

reconstruction with the angle sorting method takes 23.6 min and thus approximately as long as the orthogonality method takes with 126 directional vectors.

The execution times of each iteration and number of directional vectors include a significant amount of overhead, as it was mentioned further above. Since the overhead is expected to be constant for every execution of the reconstruction algorithm. Regardless of the overhead these results still show the considerable influence of the assigning algorithm on the overall performance.

During the performance evaluation certain iterations took an unusual amount of time to finish and led to some outliers in the computation time for the sorting algorithm. Those were non-reproducible for the particular number of directional vectors for when they occurred but were kept in the graph for reasons of consistency. One possible explanation might be the accidental use of the same GPUs by another process or other occurrences that might have led to the simultaneous occupation of servers resource and thus a decreased performance of the reconstruction algorithm since multiple people have access to it. This should be investigated in the future.

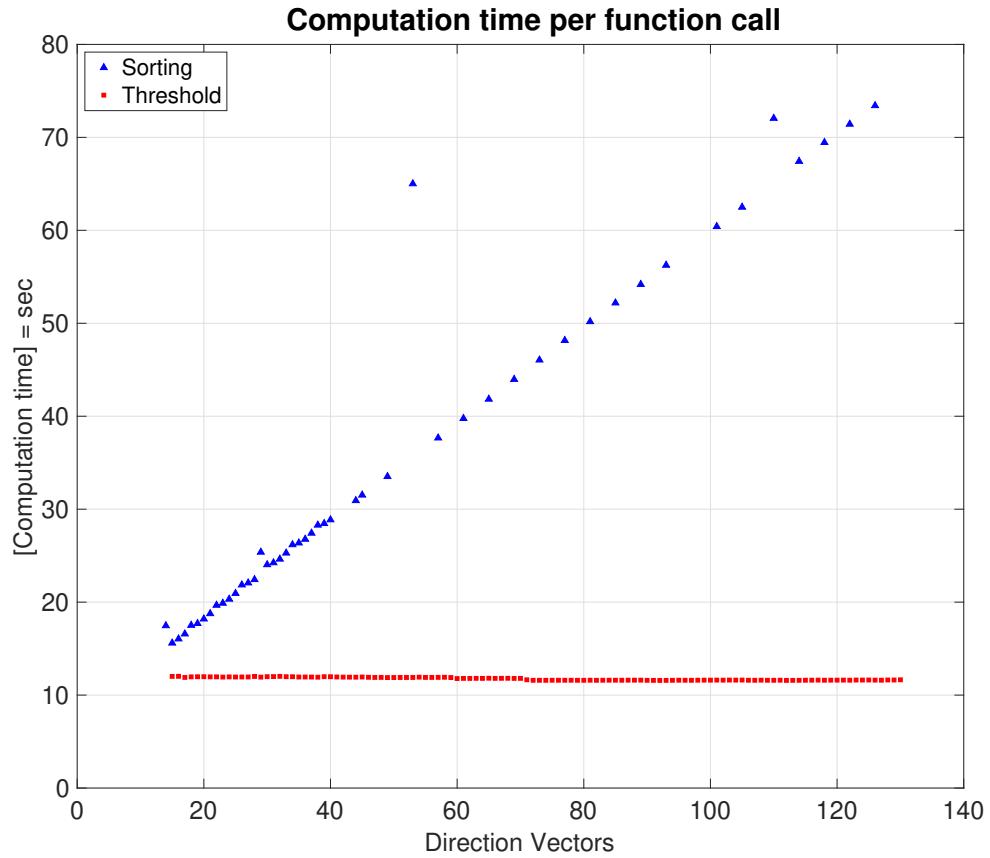


Figure 4.14: The computation time of each individual function call in sec. The execution time was divided by the number of input vectors to quantify the performance for one function call.

In Figure 4.14 the execution time for only one iteration of the reconstruction is shown. The complete computation time was divided by the number of input directional vectors to approximate the execution duration for only one function call. The resulting plot shows that the execution time for the orthogonality threshold method is rather constant for an increasing number of directional vectors. In contrast the execution time for each function call with the implementation of the sorting algorithm takes an increasing amount of time to finish. The outliers in the set of blue markers are visible again in this plot. The red markers for the threshold method seem to have a step at about 70 input vectors. The reason for that most likely is the different time of day when the executions were started. They were not done consecutively but sometimes had to be restarted at a later point. This also was the case for the performance analysis for the threshold method. Beginning with 70 input directional vectors the profiling script was restarted the day after. The reason for the small difference in execution time again might be found in the shared resources on the server and the non-constant load during the calculations.

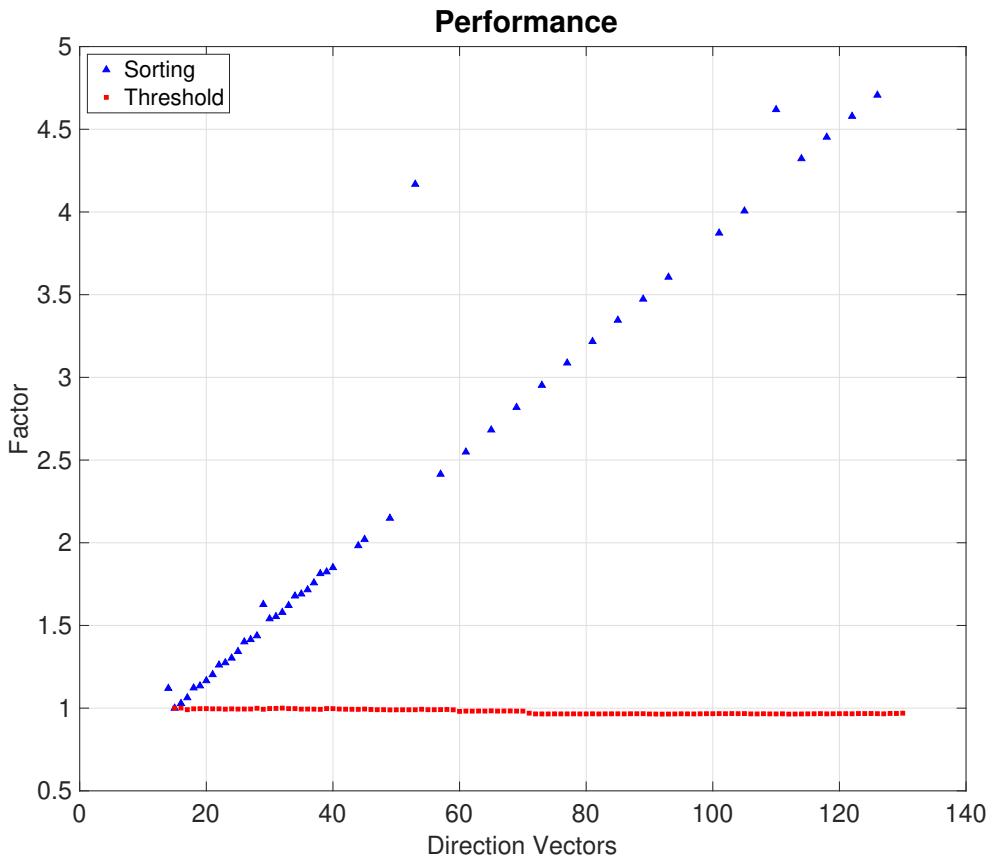


Figure 4.15: The computation time of each individual function call was normalised on the 2nd value of both methods.

To quantify the differences in performance between both approaches in Figure 4.15 the execution times for both methods were normalised on their respective second value. The second value was preferred over the first one since the first iteration always took a bit longer than the following one. A reason for that might be initial behaviour of the servers processing unit e.g. the optimisation of the cache management at the beginning of the calculations. Nevertheless, in both cases the markers for 14 directional vectors begin at the factor 1 and increase over the following iterations with more input vectors. At least this is the case for the angle sorting method. The markers for the orthogonality threshold stay rather constant at 1 whereas the blue markers end up at a factor of 4.706 for 126 input vectors. Here again the small step around the 70 input vectors arise from the different times of the start of the analysis.

4.5 Evaluation of the differentiation of tissue by directional information

The following results all were created with the orthogonality threshold method from section 3.5.3 since this method has proven to be much more computationally efficient (see section 4.4) as well as to yield an approximation of the results from the angle search method. The volume was restricted to minimise the computation time and Figure 4.2 shows the location of the reconstructed volume. For these results 35 directional vectors were created with the arbitrary segmentation approach. Per TAS one sender and nine receivers were selected. To reduce the computation time only four of the ten available aperture positions were included. The data set from the setup in section 4.1 was used. The choice of a simple setup of one small test object was preferred over complex clinical data. As a next step this procedure could be tested on simulated data where a ground truth about reflection properties of different test objects were known.

In section 4.5 the 5D-over-4D-representation of different voxels inside of the 5D data was introduced. This representation will now be used to analyse the characteristics of different tissue types (i.e. materials of the olive and the gelatin). The 5D-over-4D-representations for the three different test voxels is plotted in Figure 4.16.

In this image we see all the 4D-5D voxel values for the reconstructed data. As a reminder: the 5th dimension comprises directional information for the direction from the voxel to the emitters whereas the 4th dimension contains the information for the voxel to the receiver direction. With that in mind it becomes clear that the directional vectors 30, 31, 32, 33, 34 and 35 are pointing out of the imaging aperture and neither any emitter nor any receiver is located in this direction. Therefore, neither of the four 5D-4D-representations hold any directional information for those directional vectors.

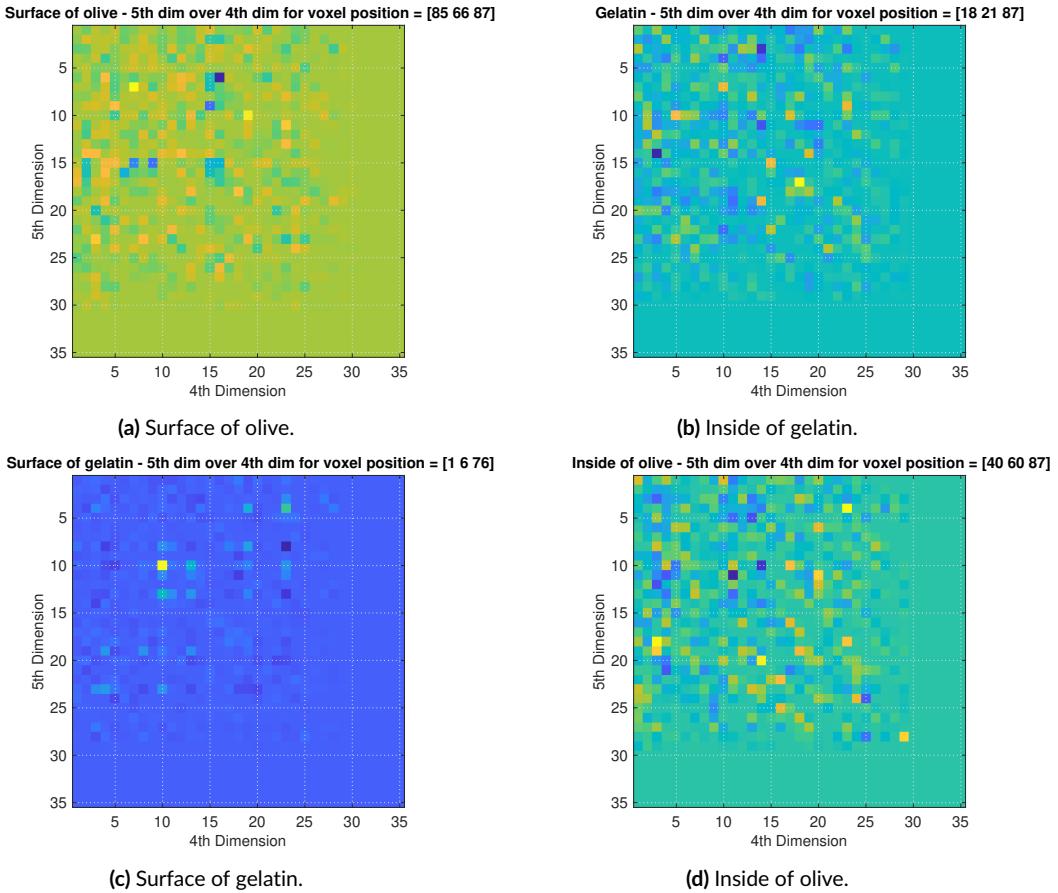


Figure 4.16: 5D-over-4D representation of three different voxels in the reconstructed image. It is assumed that the olive skin and the surface of the gelatin block both have a prevailing specular reflection characteristic whereas the gelatin and the pulp of the olive are assumed to have predominantly diffuse properties.

In Figure 4.16 the voxel values are given in reference to the indices of the directional vectors that were used. For the quantitative analysis of these results the directional vectors first have to be organised in reference to their angular relationship to each other. Section 4.6 introduces the so called graticule which represents the result in a physical context.

Since these 5D-over-4D results show features which can be used for the determination of the reflection characteristics they shall be explained qualitatively in the following.

In Figure 4.16a we see the 5D-over-4D-representation for the voxel located at the surface of the olive. This plot can be used to make a first assertion about the tissue characteristics of the examined sample. The skin of the olive is assumed to have specular reflectivity properties since it has an even surface. The surface of the gelatin block is also very even and therefore specular properties can be assumed. The 5D-over-4D representation for the surface of the gelatin can be seen in Figure 4.16c.

Figure 4.16d and 4.16b show the voxel values for the inside of the olive and the gelatin block. Since these two samples are not located at a surface, no specular behaviour is expected.

The results for the specular behaviour are shown on the left side of Figure 4.16 whereas the plots on the right side are for the diffuse cases.

In this side by side comparison the distribution is noticeable. For the specular cases on the left there are some isolated high amplitude values surrounded by comparably low values. The diffuse cases on the right appear more evenly distributed and unorganised.

In case of the surface of the gelatin block in Figure 4.16c a symmetrical distribution of the values around the leading diagonal of the voxel matrix can be observed. This phenomenon can also be seen in Figure 4.16a for the surface of the olive. In that case there are four dark values (e.g. negative amplitude) which appear to be mirrored on the leading diagonal.

The results for the diffuse case show a much more random distribution of the values and no symmetrical distribution is recognisable.

The features of the 5D-over-4D representation can be used to extract information about the surface texture of different materials. One possibility for that would be the calculation of the deviation of all voxel values per 5D-4D representation and write them into the 3D volume. The results for that are shown in section 4.7.

4.6 Angular relation between directional information

Until now the directional vectors could not be compared to each other properly. Their index was based on the order of their generation the angular relation between them was missing. A solution for this problem is presented in this section. A so called graticule is introduced. It is a representation of the azimuth and elevation of a spherical coordinate system in a two dimensional coordinate system. A simple example is shown in the following Figure:

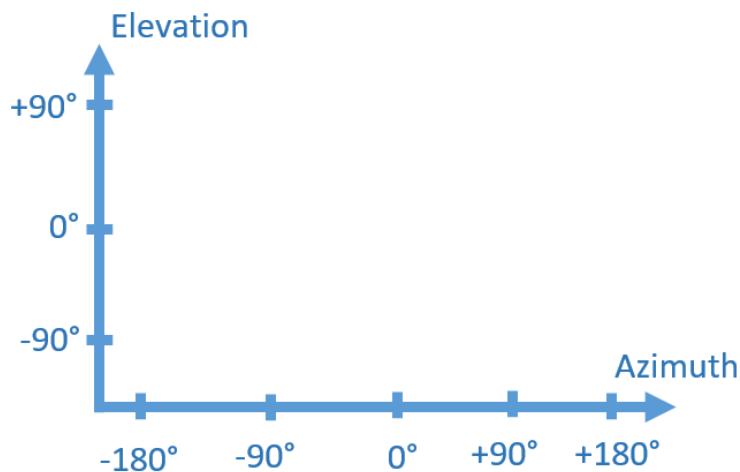


Figure 4.17: Basic representation of the graticule. The x-axis holds the information of the azimuth coordinate and the y-axis the information of the elevation.

With this representation the directional information of the single directional vectors now can be brought into a representative form so that the angular relationship between the data becomes visible. The 35 directional vectors that are used for the 5D reconstruction are shown in the Figure 4.18.

To transfer these directional vectors in the graticule the azimuth as well as the elevation is calculated for every direction vector (compare A.2). The resulting spherical coordinates are then plotted into the graticule. The values for the azimuth are plotted on the x-axis and the values for the elevation are assigned to the y-axis of the graticule. The result can be seen in Figure 4.19.

Directional vectors for the graticule

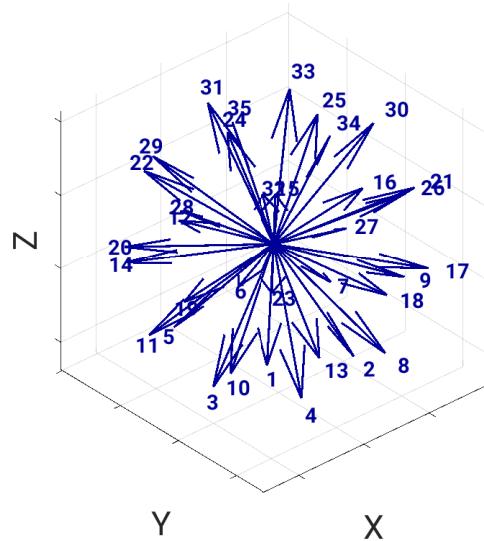


Figure 4.18: The 35 directional vectors that shall be projected into the graticule.

Graticule with 35 directional vectors

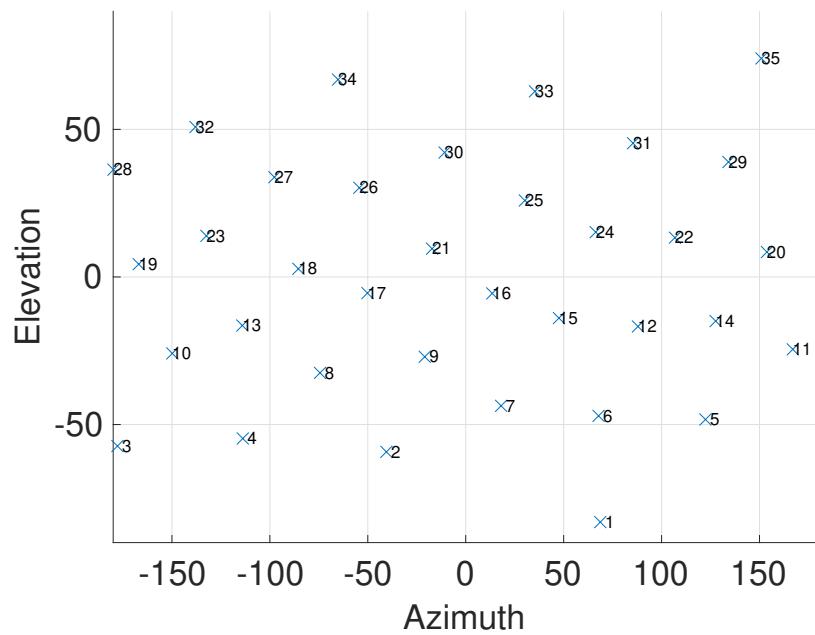


Figure 4.19: The 35 directional vectors now presented in the graticule. The units are given in degrees.

From the representation in Figure 4.19 now the directional vectors can be regarded with their angular relation to each other. This shall be shown in the following example. A test voxel is defined with the voxel coordinates [70, 82, 91]. The coordinates for the test voxel are plotted in Figure 4.20.

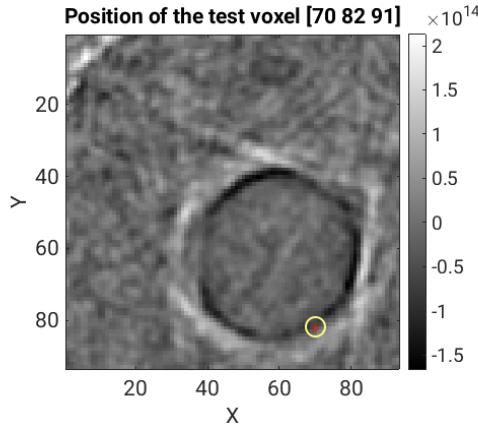
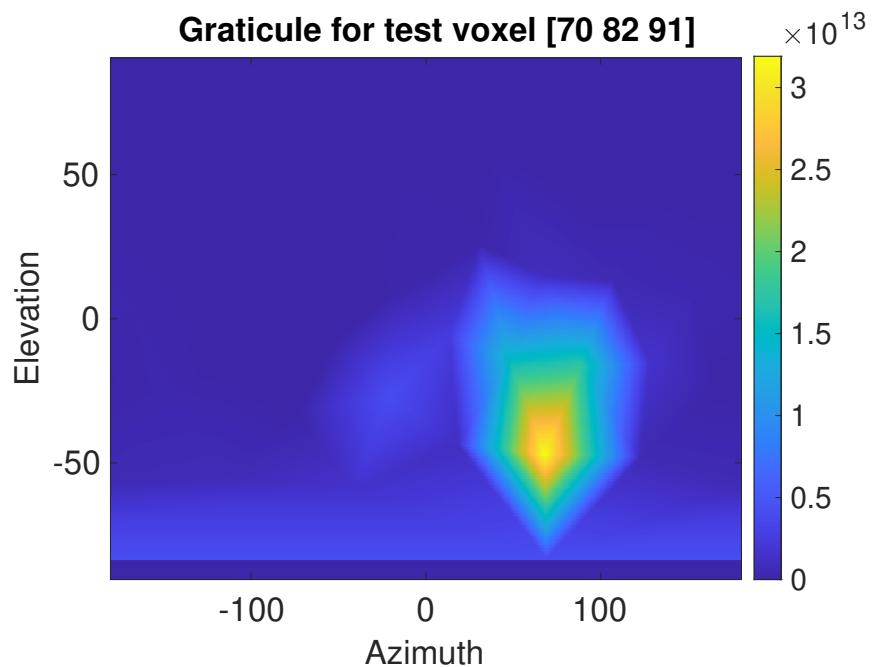


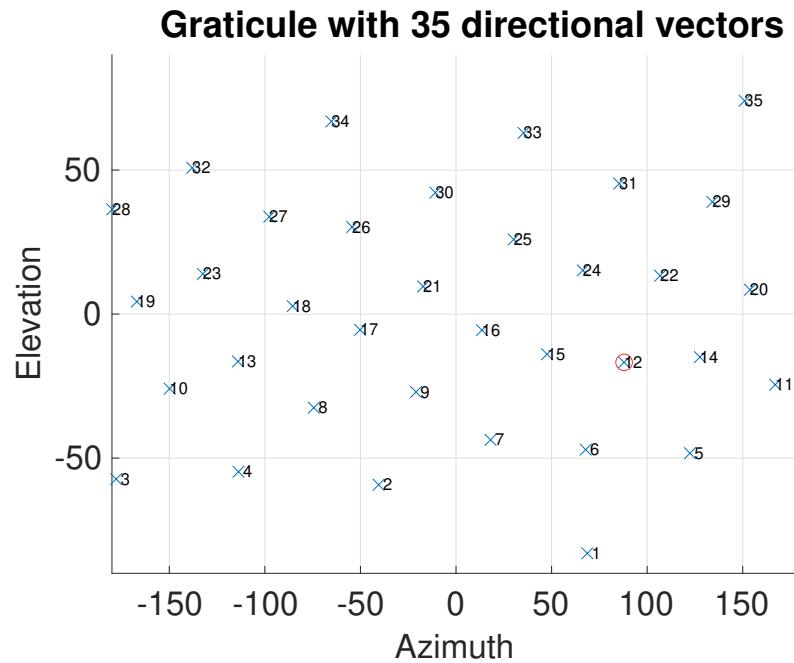
Figure 4.20: Location of the test voxel [70, 82, 91] for the graticule.

The test voxel was chosen on the surface of the olive. This surface is assumed to have specular properties as first results in section 4.5 indicated.

For the generation of the following results the data set from section 4.1 were used. The reconstruction of the 5D image was performed with the orthogonality threshold method and with 35 directional vectors. The graticule with image data can be seen in Figure 4.21.



(a) Graticule with the values of the receiver data for all directional vectors which have their origin at the emitter direction 12.



(b) Graticule of the positions of the directional vectors. Vector 12 is marked red because the data of the graticule are used from the emitter direction 12.

Figure 4.21: Graticule.

For these results, the emitter direction was exemplary chosen as 12. This means, that only the data is considered that was emitted from an emitter TAS that lays inside of the decision cone of directional vector 12. Furthermore, the receiver values for every direction are regarded. These voxel values then are placed into the graticule at the position of their corresponding receiver vectors and two dimensionally interpolated to create the image that we can see in Figure 4.21a. The colours of the figure represent the voxel value at that particular direction vector. The absolute value of the voxel values was chosen so that both positive and negative values of the A-scans are regarded. A bright structure is visible located at the position of the directional voxel 12. The structure results from the interpolation of the different voxel values located at the directional vectors surrounding vector 12. From the concentrated distribution of voxel values around the directional vector 12 the result can be interpreted as the result of a reflection on the surface of the olive with high specular properties. It remains to be examined how this structure changes if the test voxel is placed into other materials with different reflection properties and if the distribution of voxel values really allows for the classification of surface materials.

4.7 Deviation imaging & Maximum imaging

Since the final image of the reconstruction contains five dimensions the results are not easy to display in an comprehensible way. One option of displaying all the data of the five dimensions in one 3D volume was shown in Figure 4.11. In this case the voxel values were averaged along the 4th dimension and the 5th dimension. This procedure results in a 3D image which is similar to the original USCT-images without any directional information.

Another option for the representation of the data is the **deviation image**. It can be calculated from the 5D-over-4D representations and results in a three dimensional image. For each of the 5D-over-4D-images a standard deviation can be calculated. The results from section 4.5 showed a distinct pattern for tissue types with specular properties compared to tissue types with diffuse behaviour. For the specular kind there were some high peaks symmetrically distributed around the leading diagonal of the image whereas the diffuse tissue showed a more even distribution of values over the whole are of the image. With these observations in mind the standard deviation of a the 5D-over-4D characteristics for a specular scatterer should result in a higher value compared to the even distribution of the 5D-over-4D plot for the diffuse case. In the following Figure the Deviation image is shown next to the summarised image of the data set.

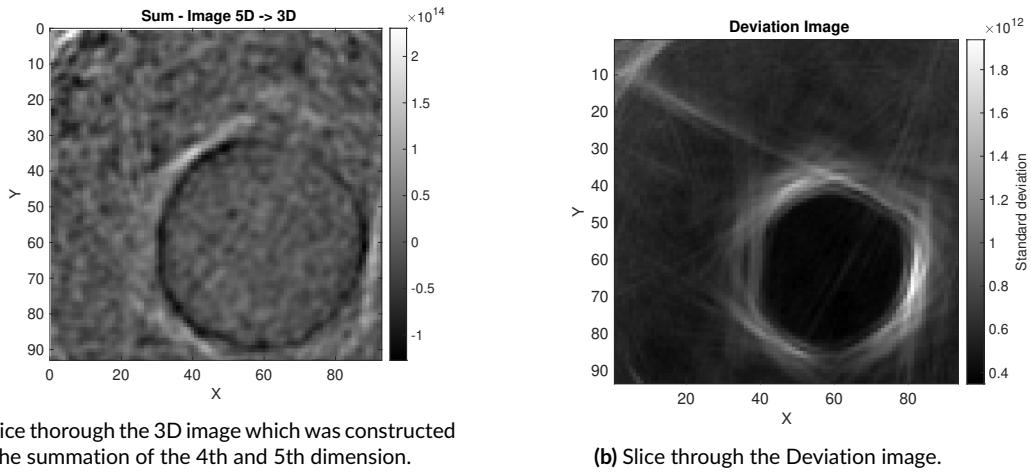
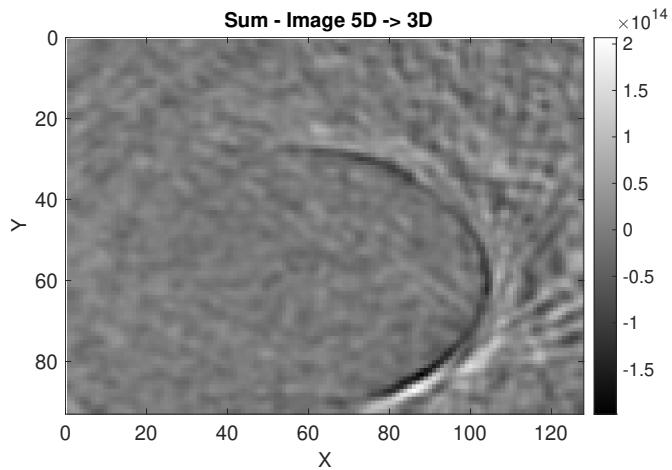


Figure 4.22: Side by side comparison of the summarised 3D image to the Deviation image.

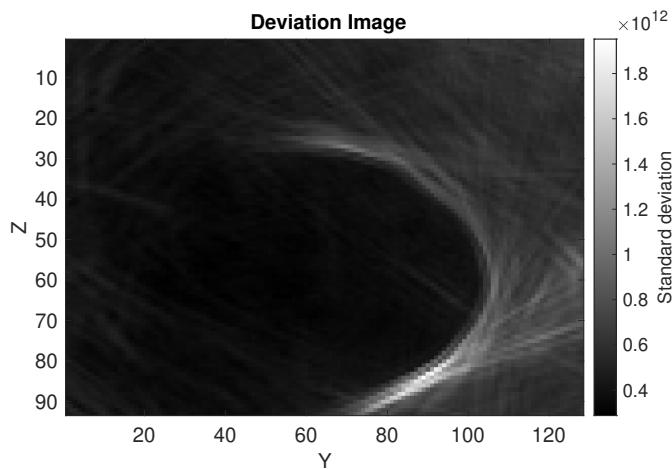
Compared to the sum-image on the left, the deviation image in Figure 4.22b shows a much higher contrast. The olive in the middle as well as the surface of the gelatin block in the top left corner are clearly visible. The speckle artefacts were suppressed and the background value is close to zero.

In section 4.5 the data for four different test voxels with different reflection properties were shown. The deviation image shows that the voxel values for the voxels that were located on the surface of the olive and on the surface of the gelatin block have a high standard deviation (e.g. high brightness in image). These test voxel that were situated in the gelatin and on the inside of the olive a much lower deviation can be observed.

With this method it could be shown that materials with a high diffuse reflectivity can be separated from low and diffuse reflecting materials. The voxels with high variance are highlighted in the deviation images and are easily distinguishable from the voxels with diffuse properties which appear much darker.



(a) Summation image.



(b) Deviation image.

Figure 4.23: Comparison of the Deviation image and the summation image. Sliced in x-direction. The oval in the middle of the image is the olive.

In Figure 4.23 the deviation image is shown from the side of the olive. This time the slices were made through the y-z-plain. Again, the edges of the olive are visible with higher contrast compared to the the summation image in Figure 4.23a.

Using the **maximum values** of each of the 5D-4D-images leads to an alternative representation of the data. For the generation of the max image the highest peak of the 5D-4D-data is saved at the position of the corresponding voxel in the 3D image. The resulting max image is shown in the following figure 4.24:

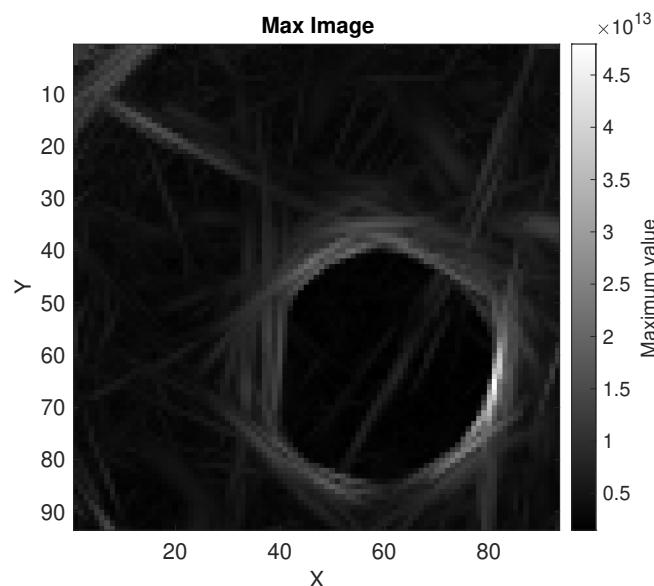


Figure 4.24: Maximum image as a representation of the five dimensional data in 3D.

Figure 4.24 shows the maximum image with the same sectioning as the deviation image in Figure 4.22b. The results show the olive in the middle of the gelatin block as highlighted structure. In the specular case the ultrasonic pulses are reflected directly from the emitter direction into the receivers that lay in the same direction. Therefore, the pulses travel only a short distance and have little interaction with surrounding matter. For these cases the reflections cause a higher amplitude in the A-scan compared to a diffuse scattering where a big part of the energy is scattered into other directions. Therefore, the maximum imaging technique also leads to a highlighted representation of mainly specular tissue.

4.8 Influence of the speed of sound correction

In this section the influence of the SOS correction on the images shall be shown. For the following results 14 directional vectors were used during the reconstruction. For the first comparison the sum images of the 5D reconstruction can be seen.

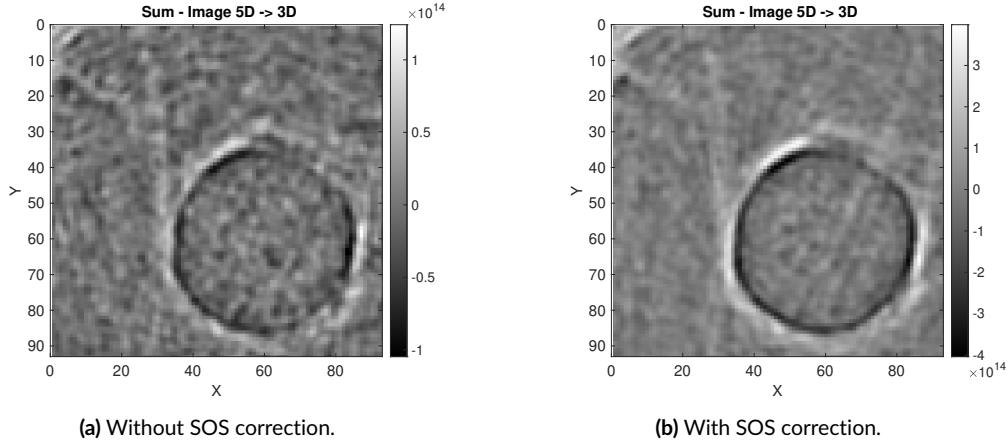


Figure 4.25: Influence of the SOS correction on the summarised image of 5D reconstruction.

It is visible on the first glance that the influence of the artefacts could be reduced. The reason for the lower visibility of the noise is the higher amplitude of the voxel values where actual image information is stored. The Figure 4.25b shows a nearly two times higher amplitude compared to the image on the left. Therefore, the impact of the artefact could be reduced with the introduction of the SOS correction.

The 5D-over-4D images for the four different materials mentioned in section 4.1 are presented in the Figure 4.26 and 4.27.

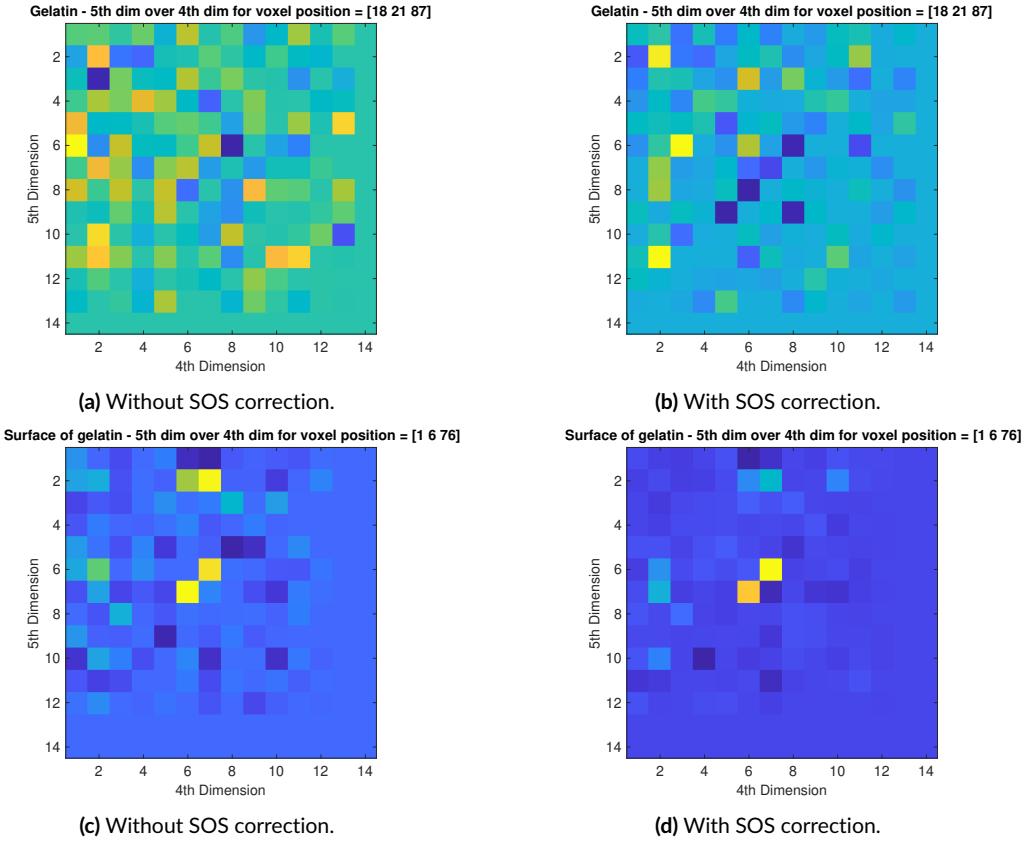


Figure 4.26: Influence of the SOS correction on the 5D-over-4D-representation of reconstructed image with 14 directional vectors. The images on the left are reconstructed without the SOS correction. The results on the right are generated while considering the speed of sound.

These results are only compared qualitatively. It was mentioned in the section 4.5 that it is assumed that tissue types or materials with specular reflection characteristics have a symmetrical 5D-over-4D representation for that particular voxel. The symmetry can be enhanced by applying the SOS correction. The peak values are increased so that the artefacts have a smaller influence. This makes them more prominent in the 4D-over-5D representation and brings out the symmetry even more.

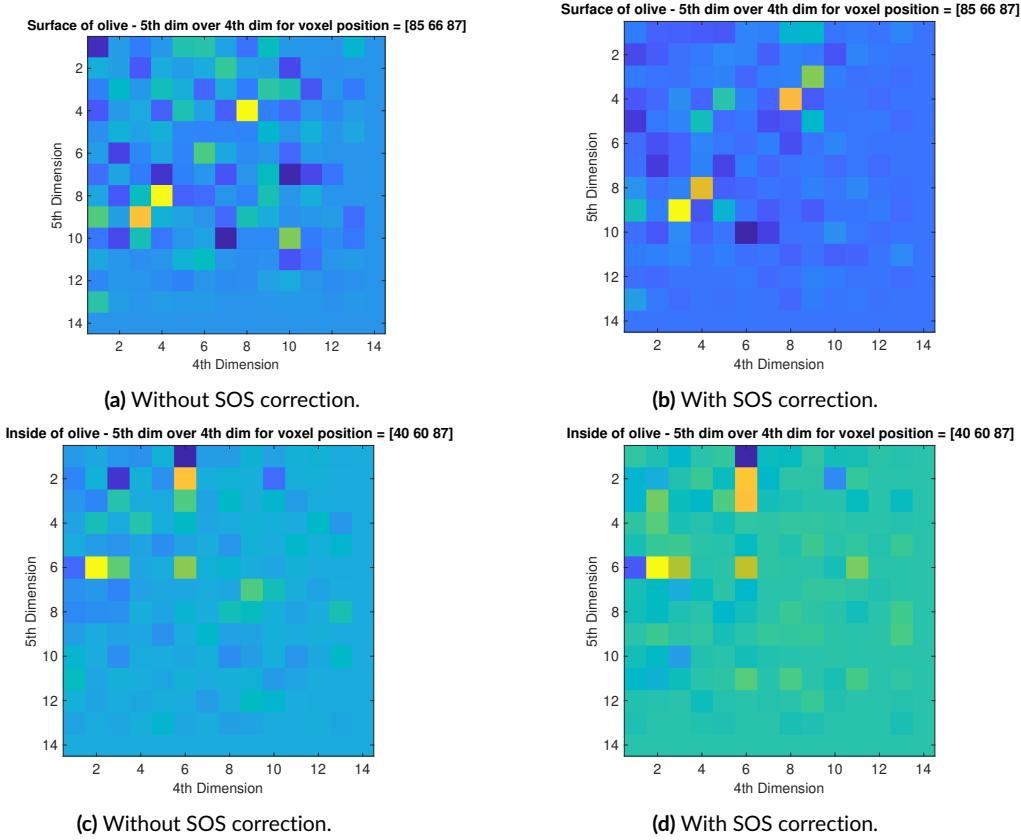


Figure 4.27: Influence of the SOS correction on the 5D-over-4D-representation of reconstructed image with 14 directional vectors. The images on the left are reconstructed without the SOS correction. The results on the right are generated while considering the speed of sound.

The biggest effect on the symmetry enhancement can be observed in Figures 4.27a and 4.27b for the skin of the olive. The result on the left shows first signs of a symmetrical distribution of the voxel values but has a lot of other superimposing components. The underlying noise could successfully be compensated by the application of the SOS correction and the symmetry could be brought about.

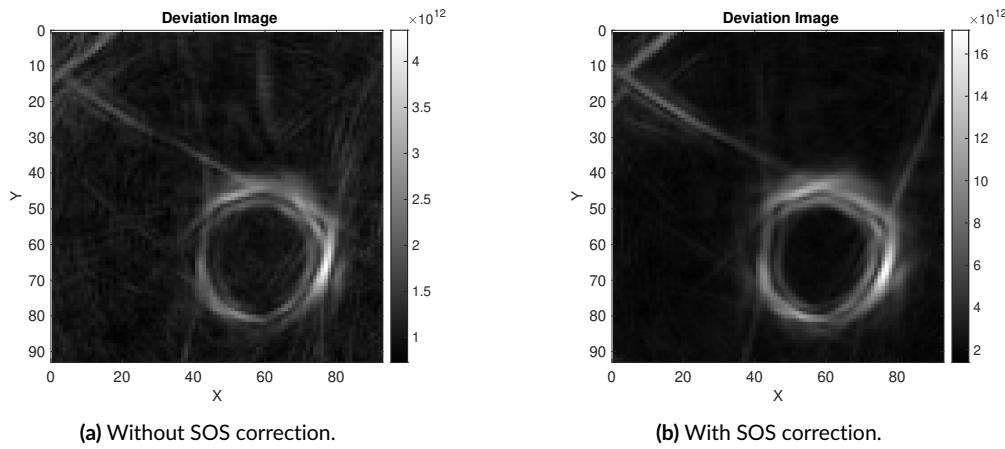


Figure 4.28: Influence of the SOS correction on the deviation image.

The results for the deviation imaging were presented in section 4.7. This imaging method benefits also from the SOS correction. The comparison of the results with and without SOS correction are shown in Figure 4.28. The contrast could be enhanced and the influence of noise reduced.

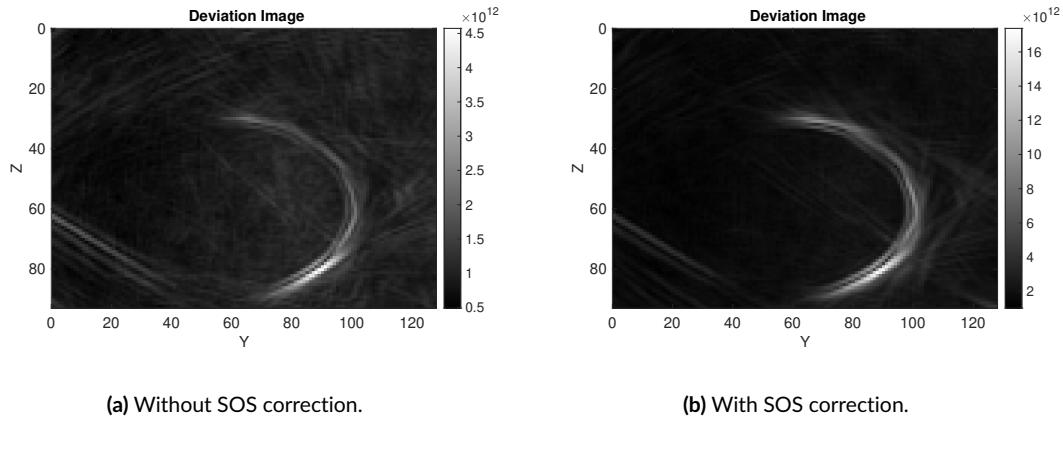


Figure 4.29: Influence of the SOS correction on the Deviation image.

Discussion & Outlook

5.1 Assessment of four and five dimensional approach

During the reconstruction of an USCT image all the directional information of the transducers are lost. This is why Patrick Hucker [7] introduced a further dimension to his reconstructed images. The 4th dimension can hold the directional information about the receiver directions or the emitter directions.

Why the 4th dimension alone is not enough to characterise the reflection properties of different tissue types can be explained with a simple experiment:

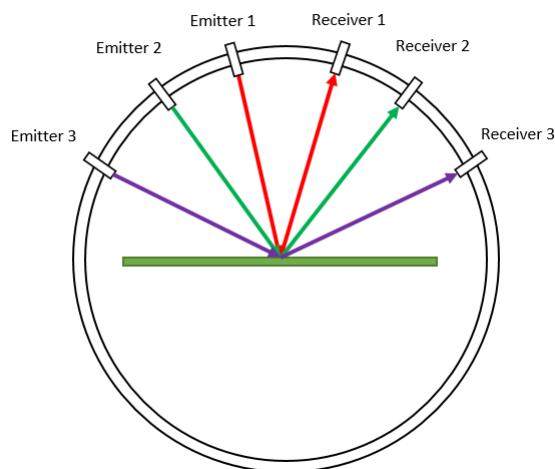


Figure 5.1: Experiment to prove that the 4th dimensional approach is insufficient for analysing the reflection characteristics of a tissue.

In Figure 5.1 the aperture of the USCT device is shown. There are three emitters and three receivers. The green structure in the middle of the aperture is one big voxel which has an even surface and optimal specular reflection properties.

Because of the specularity of the surface, receiver 1 can only detect a pulse that originated from emitter 1. Analogously, receiver 2 can only detect the pulses from emitter 2 and receiver 3 can only receive a signal from emitter 3.

For the experiment three measurements are performed. For the first measurement only the first emitter is active and emits a pulse into the aperture. For the second measurement only the second and for the third measurement only the third emitter emits. During each of the three measurements all three receivers are active and measure the pressure.

The recorded values are saved in a four dimensional image volume which is represented by the table in Figure 5.2:

Receiver/ Emitter	Receiver 1	Receiver 2	Receiver 3
Emitter 1,2,3	$\underbrace{1+0+0}_{=1}$	$\underbrace{0+1+0}_{=1}$	$\underbrace{0+0+1}_{=1}$

Figure 5.2: Measured values for four dimensional direction information.

In the table there is a column for each receiver but only one row for the emitters. This is why emitters 1,2 and 3 are summarised in the one row.

During the first measurement receiver 1 is the only receiver that detected a signal. For this measurement receiver 1 gets a red one. The remaining receivers get a red zero. The same goes for the second measurement: Receiver 2 receives a signal (green one in table), receivers 1 and 3 do not (green zeros in table). Analogously, for the third measurement only the third receiver is assigned the purple one whereas the other receivers are assigned the purple zero.

After the last measurement, the sum of the recorded signals result in a one in each receiver column. The information about what emitter was the origin of which signal is lost. If we would try to make an assumption about the reflection characteristics of this particular voxel, we would probably assume that the voxel reflected uniformly in all directions and therefore has to have a diffuse reflection characteristic.

Alternatively, we could have saved the measurement results into a 5th dimension. This is represented by the table in Figure 5.3:

Receiver /Emitter	Receiver 1	Receiver 2	Receiver 3
Emitter 1	1	0	0
Emitter 2	0	1	0
Emitter 3	0	0	1

Figure 5.3: Measured values for five dimensional direction information.

From this table we can reconstruct the emitter receiver configuration under which a signal was detected. Since only the leading diagonal of the table is occupied it becomes clear that the voxel has to have a specular reflection characteristic.

For the classification of a reflection characteristic it is therefore inevitable to analyse both the emitter and the receiver directions of a voxel. These information can only be preserved if a 5th dimension is introduced to the reconstructed image. During this thesis the previous implementation of Patrick Hücker was extended and the tools for the analysis of the reflection characteristics. For the five dimensional approach the information of original 3D image is distributed into all the sub images for each 4D-5D combination. To compensate the reduced contrast of the resulting sub images a suitable visualisation technique is presented (compare section 5.4).

5.2 Resolution of the discretisation of direction

The resolution of the discretisation has to be considered for the characterisation of the reflective properties of a tissue. This can be explained with an example in Figure 5.4. The example shows two bell curves that represent two different reflection characteristics of a tissue. The wider curve represents a mixed reflectivity with specular and diffuse parts. The narrow green curve represents a near specular scatterer. The goal is to distinguish between both curves. In the sub-figures there are three different levels of the resolution of the directional information shown.

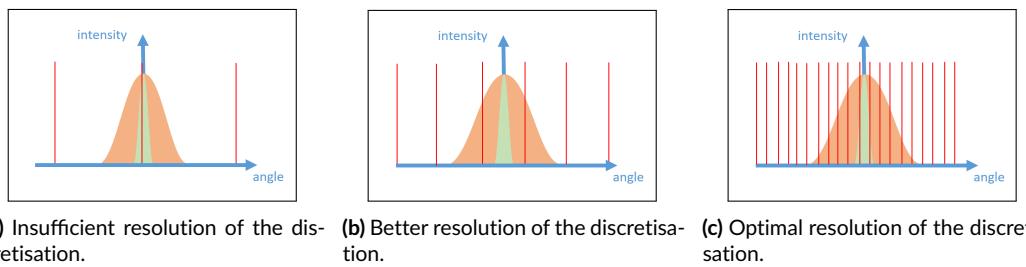


Figure 5.4: Reflection characteristics of different surfaces and angle of incidence.

In Figure 5.4a the resolution of the directional information simply is not high enough to distinguish between both curves since they are both too narrow. The second solution shows a higher resolution of the discretisation of the directional information. The red lines are much closer to each other and the orange reflection characteristics can be analysed. To distinguish between the green and the orange curve the resolution is still insufficient. Figure 5.4c shows a near optimal solution for the problem. Both reflection characteristics could be distinguished. The introduction of the arbitrary directional segmentation allows to increase the resolution of the discretisation as desired and with that to increase the resolution of the segmentation. This is one of the main features that has been introduced during the thesis and allows the analysis of the different reflection characteristics. One of the challenges of the enhancement of the resolution of the segmentation is the increasing computation time. This issue was addressed by the introduction of a more efficient assignment method.

5.3 Comparison of the assignment algorithms

In section 3.5 two approach for the assignment of the directional information were presented. The first method is the angle sorting approach which was introduced by Patrick Hucker. It was extended from 4D to 5D during this thesis. The second method is the orthogonality threshold which was developed during this thesis. The main difference between both methods are the different decision criteria and the execution time. In section 3.5 the different decision

areas of both approaches were presented in detail. In the following the three decision criteria are compared:

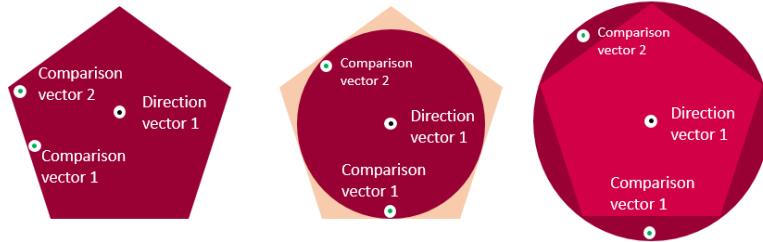


Figure 5.5: Different decision areas of the angle sorting method (left) and the orthogonality threshold with enclosed decision area (middle) and with external decision area.

For the **angle sorting** method there are fixed boundaries which correlate to the geometry of the configuration of directional vectors. For the 5D angle sorting method two different comparison vectors are assigned to the same directional vector. This can be seen in Figure 5.5 on the left side. The comparison vector 1 is located at the boundary of the pentagon. Comparison vector 2 is located in one of the 'tips' of the pentagon where two boundaries meet. The distance from comparison vector 1 to the directional vector in the middle of the pentagon is smaller than the distance between comparison vector 2 and the directional vector. Still, both comparison vectors are assigned to the same directional vector. The decision area for the orthogonality threshold method is shown on the right side. The comparison vector 1 and 2 are both located on the boundary of the decision area. Both have the same distance to the directional vector 1 in the middle of the circle.

Two comparison vectors that are located at the boundary of the same decision area can have a different angle to the comparison vector. For the angle sorting method both direction vectors may be assigned to the same directional vector. This leads to the conclusion that the angle sorting method has a varying quality of assignment. It can not be guaranteed that two comparison vectors that have the same angle to same directional vector are both assigned to the directional vector.

The **orthogonality threshold** method treats all directional vectors equally and guarantees a constant quality of the assignment. If the threshold is chosen as the enclosed circle of the pentagon (compare Figure 5.5 (middle)) some A-scans may be not regarded which results in a lower contrast of the image. On the other hand, this method leads to a distinct assignment. The other possibility is to chose the decision areas to be overlapping. An example is given in Figure 5.5 (right). For this case still a constant quality of the assignment can be assured and in addition to that all A-scans can be considered.

The other big difference between the two algorithms is their performance. While the angle sorting method iterates over every possible directional vector during the assignment process,

the orthogonality threshold method needs only are previously calculated threshold for that one set of directional vectors and can compare the comparison vectors to this threshold. That increases the performance of the algorithm considerably. Both methods have to be executed multiple times during the reconstruction. To be precise for a full reconstruction there are 1.982×10^{16} calculations necessary where the assignment methods are involved. Therefore, this big difference in performance for more directional vectors makes the angle sorting approach to a bad choice. For small sets of directional vectors the impact may be not that big. With a rising number of directional vectors the threshold orthogonality method has its advantages in performance and the quality of the assignment process.

Consequently, a trade-off has to be found between the advantages and disadvantages of the orthogonality method (uniqueness, contrast, vs. shorter computation time) and the angle search (potentially higher contrast vs. non-equal angular distribution of content and longer computation times). This thesis laid the groundwork which allows the analysis of the influence of the different decision criteria in the future with more data.

5.4 Interpretation of visualisation of the reflection characteristics

For the comprehensible visualisation of the five dimensional data the deviation imaging and maximum imaging approach was presented. For each voxel in the measurement volume the directional data was interpreted and a three dimensional representation of the data was found. From the deviation and maximum images a first distinction between the specular and diffuse behaviour of a material was made possible. For the exact determination of different tissue types further evaluations are required. In particular the deviation approach lead to a suppression of the ellipsoidal artefacts in the image. These results might be suitable as input data for a machine learning algorithm which eventually could classify different tissues. Furthermore the relatively sharp representation of the surfaces could prove helpful for the segmentation of the tissue.

5.5 Discussion of the implementation of the methods

The multidimensional reconstruction was implemented in a way that two separate images are reconstructed for each directional vector. The biggest disadvantage of this method is the computation time of the reconstruction which increases linearly with the number of direction vectors. On the other hand, during the run time of the reconstruction algorithm of the old implementation, the assignment process needed to hold the whole image volume in the memory of the GPUs. This limit was overcome by calling the reconstruction multiple times and increasing the overall reconstruction time but also keeping the memory load of

the GPU constantly low. A next step would be to transfer the prototypically implemented MATLAB code into the CUDA kernel of the reconstruction. The implementation of the reconstruction for the five dimensional images was designed to be parallelisable and it can be partitioned arbitrarily.

5.6 Conclusion & Outlook

The previous methods presented by Patrick Hücker [7] had inherent features that prevented the reconstruction of five dimensional volumes and with that made the successful classification of reflection characteristics improbable. The introduction of a 5th dimension during this thesis brought us a big step closer to the classification of tissues by analysing the reflection characteristics. The inability to increase the resolution of the segmentation of the directional information arbitrarily was addressed during this thesis. A generalisation of the problem lead to new approach which allows to arbitrarily increase the resolution of the discretisation of the directional information. This new method lay the groundwork for the detection of narrow reflection characteristics (e.g. specularity).

Moreover, the performance of the novel orthogonality threshold method was compared to the angle sorting method. The analysis concluded that the execution time per function call of the angle sorting method grew quadratically with the amount of directional vectors for the old method. This makes the old method unattractive for the assignment process with a high resolution of the discretisation. The execution time of the orthogonality method showed a linear behaviour with increasing numbers of direction vectors. This makes this method more suitable for high resolution applications. Non-overlapping decision areas were chosen for the current implementation of the orthogonality threshold to assure an unambiguous assignment to the directional vectors. The decision criteria could possibly be extended by weighting the assigned voxel values based on the orthogonality or distance to the directional vector. The influence of the different decision criteria on the resulting deviation and max images could be pursued in future.

The overall goal of this thesis was to provide insights into the reflection characteristics of different types of tissue and with these information increase the value of the reconstructed USCT-images. The distinguishing of tissue types with near specular reflection properties from diffuse scatterers could be achieved with the introduction of the deviation imaging. This insight could prove helpful in future works for recognising different breast tissues and can possibly aid in the tumour detection.

Appendix

A.1 Alternative deduction of orthogonality approach

For the analysis of the direction-dependent reflection characteristics each A-scan has to be assigned to a certain direction of the voxel. Depending on which geometry was chosen for the segmentation of the volume there is a set of normals which are orthogonal to each face of the geometry. To assign a face to each A-scan it makes sense to calculate the angle $\angle(\vec{d}, \vec{b})$ between the direction vector and each normal.

$$\angle(\vec{d}, \vec{b}) = \varphi \quad (\text{A.1})$$

$$\cos(\varphi) = \frac{(\vec{d} \cdot \vec{b})}{\|\vec{d}\|_2 \cdot \|\vec{b}\|_2} \quad (\text{A.2})$$

$$\angle(\vec{d}, \vec{b}) = \cos^{-1} \left(\frac{(\vec{d} \cdot \vec{b})}{\|\vec{d}\|_2 \cdot \|\vec{b}\|_2} \right) \quad (\text{A.3})$$

To identify the face which belongs to the corresponding A-scan the smallest angle of the set has to be found.

For every emitter-receiver combination, for each voxel and each rotation position of the aperture one calculation of the angle between two vectors has to be performed. Depending on which geometry is used for every normal of the faces of the geometry this calculation has to be repeated. The number of calculations results in:

$$\#Calculations = \#Voxel \cdot \#Emitter \cdot \#Receiver \cdot \#ApertureRotation \cdot \#Faces$$

For the case of using a 12 face dodecahedron, 628 emitters, 1413 receivers, only a single slice of 1024x64 voxels and ten aperture positions already $[628 \cdot 1413 \cdot 1024 \cdot 64 \cdot 10 \cdot 12] = 6.98 \times 10^{12}$ calculations have to be performed to find the smallest angle index. By decreasing

the complexity of these calculations the performance of the reconstruction algorithm can be greatly improved. To reduce the computational cost it is not necessary to calculate the absolute angle between each vector combination. It simply can be proven that for certain circumstances one certain combination of direction vector and face-normal has the smallest angle of the available set. The following segment is adapted from [7] and some inaccuracies were corrected.

In the following \vec{v} is the direction vector from the A-scan which should be matched best to one of the norm vectors \vec{n}_i from the N faces of the used geometry.

$$\angle(\vec{v}, \vec{n}_i) = \cos^{-1} \left(\frac{(\vec{v} \cdot \vec{n}_i)}{\|\vec{v}\|_2 \cdot \|\vec{n}_i\|_2} \right) \quad (\text{A.4})$$

For the first simplification we can neglect the \arccos in equation A.4. Since the arccosine is monotonically decreasing in the interval $[-1, 1]$ (Figure A.1) only the argument of the \arccos has to be regarded to find the smallest angle $\angle(\vec{v}, \vec{n}_i)$. To reach a small value for the arccosine function its argument has to become as big as possible.

$$\min(\angle(\vec{v}, \vec{n}_i)) = \max \left(\frac{(\vec{v} \cdot \vec{n}_i)}{\|\vec{v}\|_2 \cdot \|\vec{n}_i\|_2} \right) \quad (\text{A.5})$$

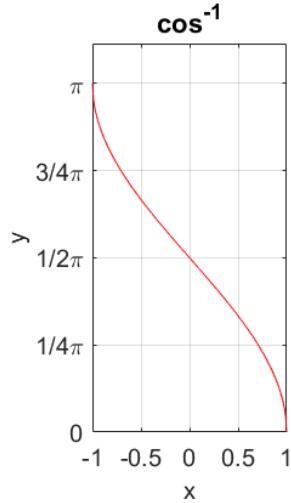


Figure A.1: Monotonically decreasing arccosine function.

Since the euclidean norm of a normal vector $\|\vec{n}_i\|_2 = 1, \forall i \in [1, \dots, N]$, the equation A.5 can be simplified further:

$$\min(\angle(\vec{v}, \vec{n}_i)) = \max \left(\frac{(\vec{v} \cdot \vec{n}_i)}{\|\vec{v}\|_2 \cdot \underbrace{\|\vec{n}_i\|_2}_{=1}} \right) = \max \left(\frac{(\vec{v} \cdot \vec{n}_i)}{\|\vec{v}\|_2} \right) \quad (\text{A.6})$$

During the search of the smallest angle between the direction vector \vec{v} and the set of normal vectors \vec{n}_i the direction vector \vec{v} does not change. The norm of the direction vector $\|\vec{v}\|_2$ is the same in every iteration of the search, thus has no influence on the final result and therefore can be removed from the formula.

$$\min(\angle(\vec{v}, \vec{n}_i)) = \max \left(\frac{(\vec{v} \cdot \vec{n}_i)}{\|\vec{v}\|_2} \right) = \max(\vec{v} \cdot \vec{n}_i) \quad (\text{A.7})$$

The final problem arises from equation A.7. The goal is to find the index i_0 for which the product of the direction vector \vec{v} and the normal \vec{n}_{i_0} is maximised:

$$i_0 = \operatorname{argmax}_{i \in 0..N} (\vec{v} \cdot \vec{n}_i) \quad (\text{A.8})$$

A.2 Spherical coordinate system

Spherical coordinates are used to create test vectors for evaluating the algorithms. Depending on the available computation power and memory it is possible to create as many normals as possible to cover as many possible directions as possible.

In three dimensions the spherical coordinate system consists of a radius r , and inclination ϑ and an azimuth φ . The conversion from one coordinate systems to the other can be seen in tables A.1 and A.2.

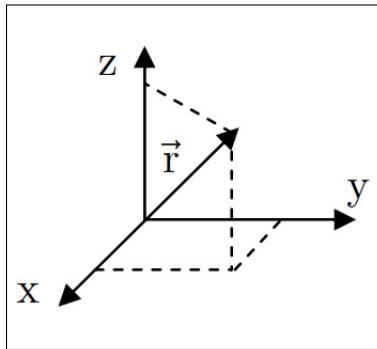


Figure A.2: Cartesian coordinate system [27].

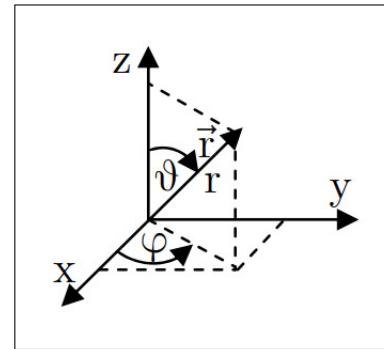


Figure A.3: Spherical coordinate system [27].

Cartesian coordinate system	Spherical coordinate system
x	$= r \cdot \sin(\vartheta) \cdot \cos(\varphi)$
y	$= r \cdot \sin(\vartheta) \cdot \sin(\varphi)$
z	$= r \cdot \cos(\vartheta)$

Table A.1: Conversion of polar coordinates to Cartesian coordinates [28].

Cartesian coordinate system	Spherical coordinate system
$\sqrt{x^2 + y^2 + z^2}$	$= r$
$\arctan\left(\frac{\sqrt{x^2 + y^2}}{z}\right)$	$= \vartheta$
$\arctan\left(\frac{y}{x}\right)$	$= \varphi$

Table A.2: Conversion of Cartesian coordinates to polar coordinates [28].

References

- [1] S. M. Lima, R. D. Kehm, K. Swett, et al., “Trends in Parity and Breast Cancer Incidence in US Women Younger Than 40 Years From 1935 to 2015,” *JAMA network open*, vol. 3, no. 3, p. e200929, 3 2020.
- [2] J. Hübner, A. Katalinic, A. Waldmann, et al., “Long-term Incidence and Mortality Trends for Breast Cancer in Germany,” *Geburtshilfe und Frauenheilkunde*, vol. 80, no. 6, pp. 611–618, 6 2020.
- [3] Robert Koch-Institut, “Bericht zum Krebsgeschehen in Deutschland 2016,” Tech. Rep., 2016.
- [4] U. Veronesi, N. Cascinelli, M. Greco, et al., “Prognosis of breast cancer patients after mastectomy and dissection of internal mammary nodes,” *Annals of Surgery*, vol. 202, no. 6, pp. 702–707, 1985.
- [5] H. G. Welch, P. C. Prorok, A. J. O’Malley, et al., “Breast-Cancer Tumor Size, Overdiagnosis, and Mammography Screening Effectiveness,” *New England Journal of Medicine*, vol. 375, no. 15, pp. 1438–1447, 10 2016.
- [6] Mayo Foundation for Medical Education and Research (MFMER), “Breast anatomy,” <https://www.mayoclinic.org/breast-anatomy/img-20007078>, 2020, [Online; accessed 1-January-2021].
- [7] Patrick Hücker, “Evaluation einer neuen Modalität für die 3D-Ultraschall-Computertomographie mit Hilfe eines Rückstreumodells,” 2014.
- [8] N. V. Ruiter, G. Göbel, L. Berger, et al., “Realization of an optimized 3D USCT,” in *Medical Imaging 2011: Ultrasonic Imaging, Tomography, and Therapy*, J. D’hooge and M. M. Doyley, Eds., vol. 7968. SPIE, 3 2011, p. 796805.
- [9] E. Kretzek and N. V. Ruiter, “GPU based 3D SAFT reconstruction including phase aberration,” in *Medical Imaging 2014: Ultrasonic Imaging and Tomography*, J. G. Bosch and M. M. Doyley, Eds., vol. 9040. SPIE, 3 2014, p. 90400W.
- [10] R. Jiřík, I. Peterlík, N. Ruiter, et al., “Sound-speed image reconstruction in sparse-aperture 3-D ultrasound transmission tomography,” *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 59, no. 2, pp. 254–264, 2 2012.
- [11] R. Dapp, M. Zapf, and N. V. Ruiter, “Geometry-independent speed of sound reconstruction for 3D USCT using apriori information,” in *IEEE International Ultrasonics Symposium, IUS*, 2011, pp. 1403–1406.
- [12] J. F. Greenleaf and R. C. Bahn, “Clinical Imaging with Transmissive Ultrasonic Computerized Tomography,” *IEEE Transactions on Biomedical Engineering*, vol. BME-28, no. 2, pp. 177–185, 1981.
- [13] J. E. Bresenham, “Algorithm for computer control of a digital plotter,” *IBM Systems Journal*, vol. 4, no. 1, pp. 25–30, 4 2010.
- [14] E. Kretzek, P. Hücker, M. Zapf, et al., “Evaluation of directional reflectivity characteristics as new modality for 3D Ultrasound Computer Tomography,” in *2015 IEEE International*

- Ultrasonics Symposium, IUS 2015.* Institute of Electrical and Electronics Engineers Inc., 11 2015.
- [15] J. H. Lambert, *Lamberts photometrie: Photometria, sive de mensura et gradibus luminis, colorum et umbrae* (1760), vol. 6. W. Engelmann, 1892.
 - [16] Han-Wei Shen - Ohio State University, “Cse 681-illumination and phong shading,” <https://web.cse.ohio-state.edu/~shen.94/681/Site/Slides.html>, 2012, [Online; accessed 7-January-2021].
 - [17] NVIDIA, “Whitepaper - nvidia’s next generation cudatm compute architecture: Fermi,” https://www.nvidia.com/content/PDF/fermi_white_papers/NVIDIA_Fermi_Compute_Architecture_Whitepaper.pdf, 2020, [Online; accessed 1-January-2021].
 - [18] Geforce rtx 3090. <https://www.nvidia.com/de-de/geforce/graphics-cards/30-series/rtx-3090/>. [Online; accessed 20-February-2021].
 - [19] Intel core i9-10900x x-series processor. <https://ark.intel.com/content/www/us/en/ark/products/198019/intel-core-i9-10900x-x-series-processor-19-25m-cache-3-70-ghz.html>. [Online; accessed 20-February-2021].
 - [20] Anuradha Karunaratna, “Cuda thread indexing,” <https://blog.usejournal.com/cuda-thread-indexing-fb9910cba084>, 2020, [Online; accessed 20-February-2021].
 - [21] NVIDIA, “Cuda c/c++ basics,” <https://www.nvidia.com/docs/IO/116711/sc11-cuda-c-basics.pdf>, 2011, [Online; accessed 20-February-2021].
 - [22] Platonic solids. https://en.wikipedia.org/wiki/Platonic_solid/. [Online; accessed 1-February-2021].
 - [23] J. Thomson, “On the structure of the atom: an investigation of the stability and periods of oscillation of a number of corpuscles arranged at equal intervals around the circumference of a circle; with application of the results to the theory of atomic structure,” *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 7, no. 39, pp. 237–265, 3 1904.
 - [24] U. Etayo, “Minimizin Riesz potentials: A way of distributing points in a spehres.”
 - [25] Anton Semechko, “Suite of functions to perform uniform sampling of a sphere.”
 - [26] C. Wittenbeck, “Konzeption, Fertigung und Evaluierung von brustähnlichen Phantomen für die 3D-Ultraschalltomographie,” 2017.
 - [27] Prof. Dr.-Ing. Gert F. Trommer, *Felder und Wellen*. Karlsruher Institut für Technologie(KIT) - Institut für Theoretische Elektrotechnik und Systemoptimierung, 10 2013.
 - [28] I. N. Bronstein, K. A. Semendjajew, G. Musiol, et al., *Taschenbuch der Mathematik*, vol. 5 ed., 2005.