

# Towards Deep Learning-based Thyroid Scintigraphy Simulation for Activity Uptake Estimation



Master Thesis  
Technical University of Munich  
School of Natural Sciences  
Chair for Computer Aided Medical Procedures

submitted by

**Korbinian Linus Träuble**

Supervisor: Prof. Dr. Nassir Navab

Advisors: Dr. Thomas Wendler, Francesca De Benetti, Christine  
Eilers

Munich, May 2023



# Abstract

Thyroid disorders affect millions of people worldwide. For this reason, a functional imaging modality called thyroid scintigraphy is used for accurate disease diagnosis and treatment planning. However, scintigraphies provide limited information about 3D functional information because they are only 2D projections. SPECT imaging provides 3D information but is too expensive and not practicable for thyroid diagnosis. The aim of this thesis is to develop an algorithm for iteratively estimating the 3D activity uptake in the thyroid gland and (if applicable) nodules using only a segmented 3D ultrasound (US), the relative pose of the US to the camera, and a 2D scintigraphy. This thesis demonstrates a proof-of-concept for this algorithm using a thyroid phantom, optical tracking, rendering methods, and Monte-Carlo (MC) simulations. Due to the unfeasible simulation time of the MC simulation step of the algorithm (with up to 44 hours per image), a deep-learning approach was introduced, yielding a time reduction of more than 99% to the order of minutes instead of days, with only minor performance losses. As 3D-US and scintigraphies are becoming the standard procedure for thyroid diagnosis, this algorithm can potentially be translated to patient data, yielding a clinical workflow to get 3D-functional information about the thyroid and if applicable nodules therein, while being cost-efficient and without the need of an additional radiation burden.



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Motivation . . . . .	1
1.2	Goal and Structure of the Thesis . . . . .	2
<b>2</b>	<b>Literature review</b>	<b>3</b>
2.1	Medical Background . . . . .	3
2.1.1	Thyroid . . . . .	3
2.1.2	Nuclear Medicine and Scintigraphy . . . . .	5
2.1.3	Ultrasound . . . . .	11
2.1.4	Computed Tomography . . . . .	15
2.2	GATE . . . . .	17
2.3	Deep Learning . . . . .	18
2.3.1	Convolutional Neural Networks . . . . .	18
2.3.2	Conditional Diffusion Models . . . . .	21
<b>3</b>	<b>Methods</b>	<b>25</b>
3.1	Approach . . . . .	25
3.2	Optical Tracking . . . . .	27
3.3	Ground Truth Generation with GATE . . . . .	29
3.4	Rendering . . . . .	34
3.5	Deep Learning . . . . .	36
3.6	Iteration . . . . .	36
<b>4</b>	<b>Experimental Setup</b>	<b>37</b>
4.1	Measurement . . . . .	37
4.1.1	3D Printing . . . . .	37
4.1.2	Tracking . . . . .	40
4.1.3	CT of Phantom . . . . .	44
4.1.4	Scintigraphy of Phantom . . . . .	45
4.1.5	Scintigraphy of Syringe . . . . .	47

4.2	Dataset for Deep Learning . . . . .	47
<b>5</b>	<b>Implementation</b>	<b>51</b>
5.1	Ground Truth Generation with GATE . . . . .	51
5.1.1	Camera Geometry . . . . .	51
5.1.2	Phantom Geometry . . . . .	52
5.1.3	Digitizer . . . . .	54
5.1.4	Time Parameters . . . . .	55
5.1.5	Physics . . . . .	55
5.1.6	Source Definition . . . . .	55
5.2	Rendering . . . . .	56
5.3	Deep Learning . . . . .	58
5.4	Iteration Loop . . . . .	61
<b>6</b>	<b>Results</b>	<b>65</b>
6.1	GATE . . . . .	65
6.1.1	Discussion . . . . .	65
6.2	Rendering . . . . .	71
6.3	Deep Learning . . . . .	73
6.4	Iteration Loop . . . . .	74
6.4.1	GATE . . . . .	74
6.4.2	Rendering + Neural Net . . . . .	79
<b>7</b>	<b>Conclusion and Outlook</b>	<b>85</b>

# Chapter 1

## Introduction

### 1.1 Motivation

Thyroid scintigraphy is a commonly used imaging technique for evaluating thyroid function and detecting thyroid disorders such as hyperthyroidism, hypothyroidism, and thyroid nodules [28]. It provides functional information about the distribution and uptake of a radioactive substance in the thyroid gland, which is crucial for accurate diagnosis and treatment planning. Thyroid disorders affect millions worldwide, and early and accurate diagnosis is essential for appropriate patient treatment [68]. Scintigraphy is also used in post-treatment monitoring of thyroid cancer patients to assess the response to radioactive iodine therapy and detect any residual or recurrent disease [44]. Conventional thyroid scintigraphies involve using a gamma camera that yields a 2D projection image. However, this provides limited information about the 3D activity distribution within the thyroid. The accuracy of thyroid scintigraphy depends on several factors such as the patient's anatomy, as well as the positioning of the gamma camera, which can lead to potential errors in activity estimation. SPECT imaging provides 3D information about the activity distribution but is more expensive than thyroid scintigraphy and it is not practicable in a clinical setting for thyroid imaging. This highlights the need for improved and cheap techniques that can provide more accurate and reliable 3D activity maps for personalized patient management. Monte-Carlo simulations are the gold standard for simulating medical images but they are very time-consuming because of their computational demand [20]. Over the last decade, deep learning (DL), a subset of machine learning, has shown great success in medical imaging applications, such as image segmentation, image registration, and image synthesis [34] [70]. Using DL in thyroid scintigraphy simulations has the potential to overcome the limitations of traditional methods and provide similar results in significantly reduced time and therefore cost.

## 1.2 Goal and Structure of the Thesis

The main goal of this thesis is to develop a deep learning-based thyroid scintigraphy simulation for iteratively estimating the 3D activity uptake in a specific patient using only a segmented 3D ultrasound (US), the relative pose of the US to the camera, and a 2D scintigraphy.

A literate review is presented in Chapter 2. This chapter focuses on the relevant medical background, including the thyroid gland, nuclear medicine, ultrasound, and computed tomography. The GATE platform framework is discussed as well, and an overview of deep learning concepts is provided. Additionally, some architectures used in the Deep Learning field, including convolutional neural networks, and conditional diffusion models, are presented. Chapter 3 covers the methodology of the project such as including optical tracking, ground truth generation with GATE, rendering, deep learning, and the iteration process. In Chapter 4, an overview of the methods used for the measurements is provided, including 3D printing, tracking, CT scans of a phantom, scintigraphies of phantoms, and scintigraphies of a syringe. The setup for Deep Learning is briefly introduced as well. Chapter 5 covers the implementations of the ground truth generation with GATE, including camera and phantom geometry, digitizer, time parameters, physics, and source definition. Moreover, it discusses the rendering and deep learning implementation, as well as the iteration loop algorithm. In Chapter 6, the results and discussion obtained from the experiments, including GATE, rendering, deep learning, and the iteration loop are elaborated. The conclusion and outlook are summarized in Chapter 7 with a focus on the main findings of this thesis. Finally, possible future directions for further research in the field are presented.

# Chapter 2

## Literature review

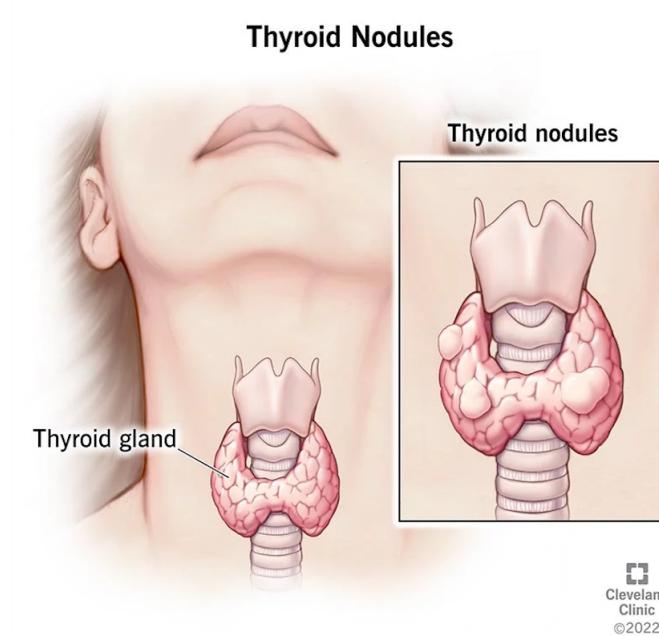
### 2.1 Medical Background

#### 2.1.1 Thyroid

The thyroid is a butterfly-shaped endocrine gland located in the neck wrapped around the trachea (see Fig. 2.1). In adults, it weighs between 15 and 20 grams. Its main purpose is to secrete the two major hormones thyroxine (T4) and tri-iodothyronine (T3). Eventually, almost all the T4 is converted into T3. Both have the function to influence the metabolic rate of the body profoundly. The thyroid hormones increase the transcription of large numbers of genes, which leads to a generalized increase in functional activity throughout the body. This affects for example growths, body weight, body temperature, and sleep.

Iodine is required to produce T4. An adult needs about 1 mg of iodine per week to form normal quantities. Iodine concentrates in the thyroid up to about 30 times its concentration in the blood, but can also rise to 250 times when it is maximally active.

The diseases of the thyroid can be categorized as hyperthyroidism, hypothyroidism, and nodules. In hyperthyroidism, the thyroid is bigger (to two to three times the normal size), and it secretes thyroid hormones at rates 5 to 15 times the normal amount. The autoimmune disease Grave's disease is the most common form of hyperthyroidism where antibodies have a stimulating effect on the thyroid. Thyroid localized adenoma (a tumor) that secrets large amounts of thyroid hormones can also result in hyperthyroidism. The symptoms of hyperthyroidism range from a high state of excitability, intolerance to heat, and extreme weight loss to nervousness or other psychic disorders. It is treated with radioactive iodine, which destroys most of the secretory cells.



*Figure 2.1: Illustration of a thyroid and thyroid nodules. [15]*

Hypothyroidism is the opposite of hyperthyroidism. It is often initiated by autoimmunity against the thyroid gland, called Hashimoto disease. In contrast to autoimmune disease in hyperthyroidism, in hypothyroidism, immunity destroys the gland rather than stimulating it. Most patients first have a thyroid inflammation called thyroiditis which causes progressive deterioration and finally fibrosis of the gland. Other types of hypothyroidism also exist, such as endemic colloid goiter caused by dietary iodine deficiency, and destruction of the thyroid by irradiation, or surgical removal of the thyroid. In all cases, the symptoms range from fatigue (sleeping up to 12 to 14 hours a day), extreme muscular sluggishness, increased body weight to scaliness of the skin, and even the development of a froglike husky voice. In severe cases of almost total lack of the thyroid hormone, patients suffer from myxedema (an edematous<sup>1</sup> appearance - see Fig. 2.2). Hypothyroidism is treated by daily oral ingestion of a tablet containing thyroxine. If properly treated, myxedema patients can live for more than 50 years, into their 90s. [25]

Thyroid nodules are defined as discrete lesions within the thyroid gland, radiologically distinct from surrounding thyroid parenchyma [16]. They are most common in women and older populations. In the United States, the annual incidence of thyroid nodules is approximately 0.1%, yielding a 10% lifetime probability for developing a thyroid nodule [66]. Women are four times more likely to have nodules, which could be explained by the hormonal influences of both estrogen and progesterone [41]. Nodules are clinically relevant because they can cause thyroid dysfunction and

<sup>1</sup>abnormally swollen with fluid



Figure 2.2: Patient with myxedema. (Courtesy Dr. Herbert Langford.) [25]

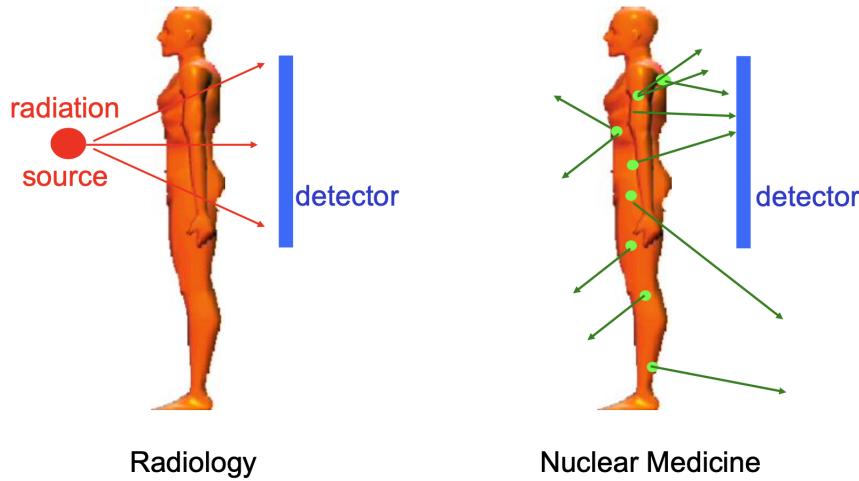
thyroid cancer. The reported prevalence of malignant nodules is low. Evaluation by biopsy reported 4.0% - 6.5% of malignant nodules independent of the nodule size [28]. Different modalities exist to evaluate whether a nodule is malignant or not. Serum Markers for example measure the serum thyroid stimulating hormone (TSH) concentration, as the risk of malignancy increases as the serum TSH increases. The most reliable diagnostic tool is a biopsy to evaluate the tissue in a laboratory. This is usually done in conjunction with ultrasound. First, to see if there are any suspicious nodules. Second, to guide during the biopsy. Other diagnostic tools include elastography, PET, and thyroid scintigraphy [49]. Scintigraphy and ultrasound will be explained in more detail in Section 2.1.2 and Section 2.1.3 respectively.

## 2.1.2 Nuclear Medicine and Scintigraphy

### Nuclear Medicine

Nuclear Medicine consists of using radioactively labeled substances, called radionuclides or tracers, to provide diagnostic information about diseases or to treat it. It differentiates itself from radiology by the fact that the source of radiation is inside the patient rather than outside (see Fig. 2.3). It relies on the tracer principle that states that radionuclides are radioactive isotopes that act chemically identical to their stable isotopes [6]. This allows “radiolabeling”, which means replacing one or more atoms in biochemical macromolecules with their radioisotope. These radioisotopes decay and emit radiation that can be detected with a detector outside the patient. In Table 2.1 you can see several tracers together with their emission energy, half-life, and use. Radioactive activity is given in the unit Becquerel (Bq), which is defined by one decay per second [74]. For example, if there is an activity of 10 Bq inside of one voxel<sup>2</sup>, there are 10 decays expected per second inside of that voxel. The power of nuclear medicine lies in sensitivity and functional information. While

<sup>2</sup>3D version of a pixel.



*Figure 2.3: In radiology the source is outside of the patient (left) while in nuclear medicine the source is within the patient (right). [77]*

Radionuclide	Half-Life [h]	Energy [keV]	Clinical use
$^{99m}\text{Tc}$	6.02	140.5	$^{99m}\text{Tc}$ pertechnetate for imaging of thyroid [18]
$^{201}\text{Tl}$	73	70, 167	$^{201}\text{Tl}$ for myocardial perfusion [7]
$^{123}\text{I}$	13	159	$^{123}\text{I}$ -sodium iodide for thyroid function [14]
$^{18}\text{F}$	1.83	511	$^{18}\text{F}$ -FDG for cancer imaging [50]

*Table 2.1: Radionuclides together with their half-life, emission energy, and clinical use. [76] [64] [71]*

radiology provides very high anatomical information, it lacks the ability to provide functional information. In nuclear medicine, there is only the need for nanomolar or picomolar concentrations to form high-quality images. This sensitivity together with the large selection of available radionuclides to target very specific biological processes allows for a very reliable diagnosis. Nevertheless, because there is a certain dose delivered during each imaging session, the doctor needs to establish careful dosimetry and evaluation of the necessity of the patient. In fact, radioisotope therapy is also a part of nuclear medicine. There, the radiation is not used for imaging, but to destroy tissue. Because it is not used in this thesis, it will not be explained further.

Once the tracer is administered and emits radiation, this radiation needs to be detected and processed into an image. The most used device capable of doing that is called a gamma camera. It is capable of detecting gamma radiation. Gamma radiation is usually within energies ranging from 80keV to 500keV (see corresponding frequencies in Fig. 2.4). There are no hard limits on the energy range as it is defined as radiation created from nuclei decays. Gamma radiation has a sufficient penetration depth to escape the body from deep-lying organs and not get attenuated significantly in the tissue. However, it is still easily shielded with lead. The

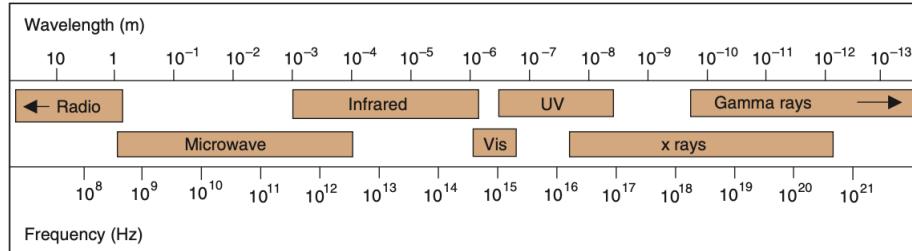


Figure 2.4: Electromagnetic spectrum with different regions. Visible light (Vis), Ultraviolet light (UV). [14]

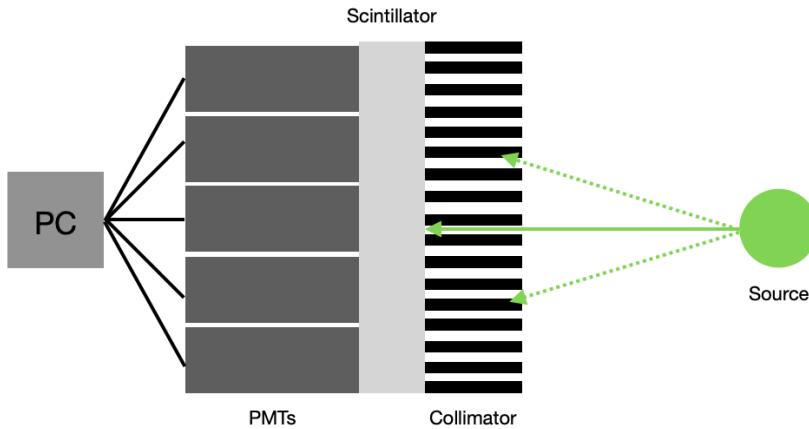


Figure 2.5: Schematic illustration of a gamma camera with the components: parallel collimator, scintillation crystal, PMTs, and readout electronics.

basic principle of a gamma camera is shown in Fig. 2.5. It always consists of a collimator, scintillation crystal, photomultiplier tubes (PMTs) and some readout electronics. Because the radiation emits in all directions from the source, a collimator is necessary to block out all nonparallel rays to get a projective image of the source distribution. Collimators are usually made out of high-absorbing materials such as lead or tungsten. Hence, in parallel-hole collimators, only rays more or less perpendicular to the scintillation crystal hit the detector. A collimator can take all kinds of shapes to act as a "lens" for the gamma rays. For example, there are diverging collimators, which minify the source, or converging collimators which magnify the source. There is an inevitable trade-off between sensitivity and resolution depending on the collimator. For example in a parallel-hole collimator, the sensitivity and resolution can be calculated by

$$\text{Sensitivity} = \frac{\sqrt{3}}{8\pi} \frac{d^2}{a_{eff}^2} \frac{d^2}{(d+t)^2} \quad (2.1)$$

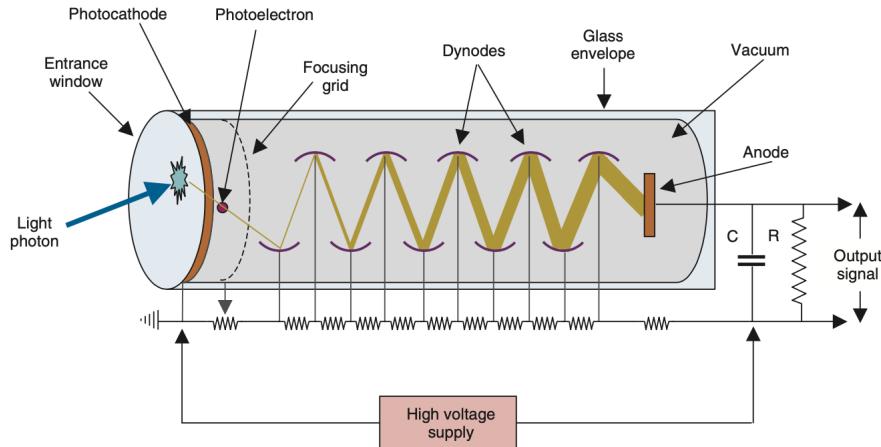


Figure 2.6: Schematic illustration of a photomultiplier tube. [14]

$$\text{Resolution} = d \frac{a + h}{a_{\text{eff}}} \quad (2.2)$$

where  $d$  is the hole diameter,  $t$  is the septal thickness, and  $h$  is the perpendicular distance from the source to the detector,  $a_{\text{eff}} = a - \frac{t}{\mu}$  where  $a$  is the hole length and  $\mu$  the attenuation coefficient [67].

The next component of the camera is the scintillation crystal. It converts deposited energy from the gamma rays into visible light. The amount of energy deposited is proportional to the quantity of light emitted. The most used material for scintillation crystals is Thallium-doped Sodium Iodide NaI(Tl). It has a light yield of 38000-415nm photons per MeV. This emitted visible light hits the PMT, which is responsible for producing a pulse of electrical current when stimulated by very low light signals. The PMTs usually have a diameter of 5cm. They consist of a photocathode that ejects photoelectrons when hit with light. Close by the photocathode is a metal plate called a dynode, which is under positive voltage relative to the photocathode. It is covered by a material having relatively high secondary emission characteristics, which emits additional electrons. This is repeated with a second dynode nearby, which has again a higher voltage relative to the first dynode and is coated with the same material. This occurs 9-12 times until an avalanche of electrons is collected at the anode, which can be measured (see Fig. 2.6). In a gamma camera, there are many PMTs arranged in an array. The position of a gamma hit in the crystal is detected by multiple PMTs and then determined by taking a weighted average or centroid of the PMT signals. Finally, an image can be formed out of the positions of the hits within the crystal. To protect the personnel in the clinic and to prevent radiation that is not from the source hitting the crystal, there is usually a lead shielding around the whole camera except the direction where the patient is. [14]

## Thyroid Scintigraphy

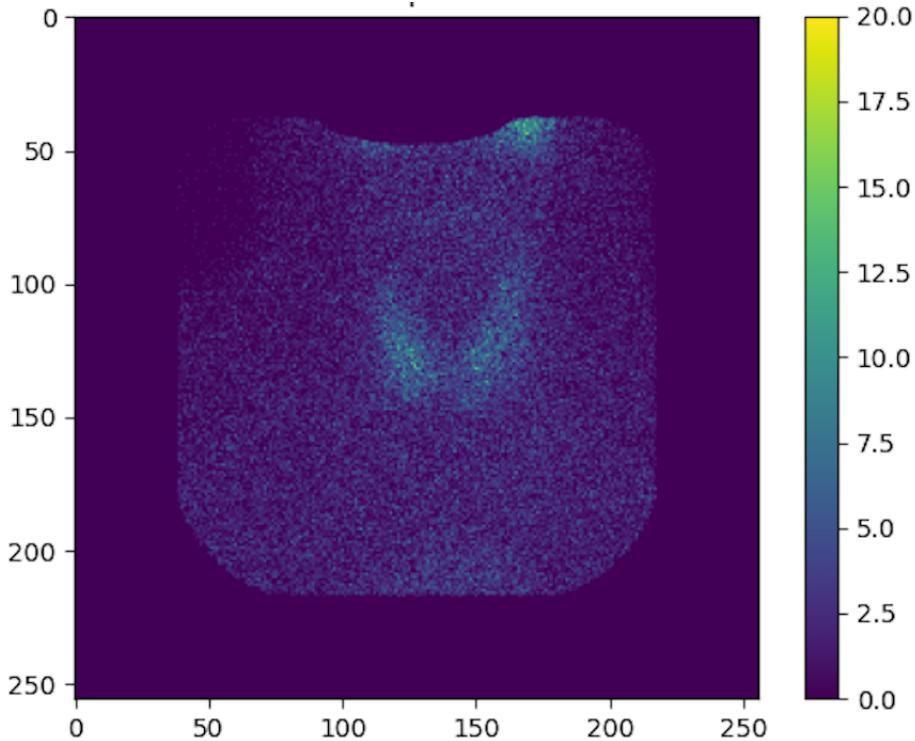
As described in Section 2.1.1, there are many different thyroid diseases with severe symptoms. In some cases, the anatomical information of the thyroid acquired for example by magnetic resonance imaging, computed tomography or ultrasound (US) does not give enough information for a reliable diagnosis. It is the function of the thyroid which is the most important observable to characterize and treat disease. Hence, when there is a suspicion of a disease, the US is followed up by a functional image of the thyroid, which is called thyroid scintigraphy. As described in Section 2.1.1, iodine is essential for the function of the thyroid. Evaluating the iodine concentration in the thyroid gives the necessary information for a reliable diagnosis. The isotope  $^{123}\text{I}$  would be a good candidate for a thyroid scan because it behaves as iodine and is radioactive. It is used frequently for thyroid scintigraphies [28]. Nonetheless,  $^{99m}\text{Tc}$ -pertechnetate behaves analog to iodine as it accumulates in the thyroid and has some advantages over  $^{123}\text{I}$  [5]. Its hands-on availability, price, excellent image quality, low radiation exposure, and short half-life are the reason many thyroid scintigraphies are performed with  $^{99m}\text{Tc}$ -pertechnetate [42]. Because  $^{99m}\text{Tc}$ -pertechnetate retains stable in the thyroid for only 30 minutes, the scan has to be taken around 20 minutes post-injection [37] [24]. The protocol for thyroid scintigraphy according to [8] is:

1. Inject 2-10 mCi of  $^{99m}\text{Tc}$ -pertechnetate intravenously.
2. Wait 15-30 min for the  $^{99m}\text{Tc}$ -pertechnetate to accumulate in the thyroid.
3. Acquire image for 5 min.

The effective dose for adults is 0.15 mSv per mCi injected assuming 15% uptake. For comparison,  $\text{Na}^{123}\text{I}$ -Iodide has an effective dose of 1.3 mSv per mCi with the same parameters [8]. An example of a thyroid scintigraphy can be seen in Fig. 2.7. The relevant metric for the assessment of hyperthyroidism and hypothyroidism is the uptake of the tracer in the thyroid. To get the counts inside the thyroid a segmentation of the thyroid in the scintigraphy is necessary. The uptake is calculated by the percentage of the number of hits inside the thyroid from the total amount of hits calibrated by the injected activity [2]:

$$Uptake[\%] = \left[ \frac{count_{thyroid} - count_{background}}{syringecount_{pre-injection} - syringecount_{post-injection}} \right] \cdot 100\% \quad (2.3)$$

In adults, the uptakes of the thyroid can be summarized in Table 2.2. Healthy patients have an uptake between 0.9% and 2.6%. For hyperthyroiditis patients, a



*Figure 2.7:* Thyroid scintigraphy of a patient. The dark background and edge are due to the shielding around the collimator. In this area, no hits arrive at the crystal.

$^{99m}\text{TcO}_4-$ (% Uptake, 30min)	Euthyroid	Hyperthyroid	Hypothyroid
Male	$1.52 \pm 0.79$	$8.19 \pm 6.24$	0.60
Female	$1.76 \pm 0.86$	$9.76 \pm 6.29$	$0.52 \pm 0.12$
Total	$1.73 \pm 0.85$	$9.33 \pm 6.24$	$0.52 \pm 0.37$

*Table 2.2:*  $^{99m}\text{Tc}$ -pertechnetate uptakes in % after 30min in n=1000 adults with euthyroid (healthy), hyperthyroidism and hypothyroidism [5].

higher uptake is expected, as more  $^{99m}\text{Tc}$ -pertechnetate accumulates in the thyroid, ranging to higher uptakes than healthy patients at 3.1% to 15.5%. The opposite holds for hypothyroidism with a range of 0.1% to 0.9%. It is also noticeable that women have a higher uptake compared to male patients. There are also many factors influencing the uptake such as female oral contraceptives, or patients previously treated for thyroid diseases [5]. As described in Section 2.1.1, thyroid nodules are very common in the population, but only very few are malignant. The evaluation of these nodules should identify patients with malignant diseases and avoid anxiety and surgical procedures for patients with benign nodules. In scintigraphies, nodules are classified by their ability to take up the tracer: non-functioning nodules are called *cold*, normally functioning nodules are called *warm*, and hyperfunctioning nodules are called *hot* [53]. In a study where all patients with nodules underwent a radionuclide scan and surgery regardless of the functional status of the nodule,

84% of the nodules were cold, 10.5% were warm and 5.5% were hot. Malignancy was found in 16% of the cold nodules, 9% in warm nodules, and 4% in hot nodules [3][4]. Hence, cold nodules have the highest probability of being malignant but are most often benign. A hot nodule reduces the likelihood of malignancy but does not exclude it. This is also the main limitation of scintigraphy for nodule evaluation because it cannot distinguish between malignant and benign, but can only be used to assign a probability of malignancy. Also, these probabilities are different for each radionuclide used. A nodule can be hot with an  $^{99m}\text{Tc}$ -pertechnetate but cold on a  $^{123}\text{I}$  scan [53].

### 2.1.3 Ultrasound

Ultrasound (US) imaging is a noninvasive, cheap, and widely available [39] imaging modality. It uses the physical principle of wave reflection and the piezo-electric effect. Sound exhibits longitudinal waves of compression and rarefaction. The frequency  $f$  of a wave is defined by

$$f = \frac{c}{\lambda} \quad (2.4)$$

where  $c$  is the speed of sound in the medium and  $\lambda$  the wavelength. The spectrum of sound can be seen in Fig. 2.8. According to the American National Standards Institute [30], every frequency higher than the human hearing range of 20-20.000Hz, is considered US. The energy absorption of sound in matter is modeled by the Beer-Lambert law

$$I(y) = I_0 e^{-\alpha y} \quad (2.5)$$

where  $y$  is the distance traveled inside a medium with attenuation coefficient  $\alpha$  which is frequency and matter dependent. As the frequency increases, the wave can get less deep into the tissue. As the wavelength decreases the resolution increases. This is again a trade-off between resolution and depth which has to be adapted to the targeted tissue [47]. For example, to image a fetus in a pregnant woman, a penetration depth of 50cm is necessary and hence a frequency of 1 MHz should be

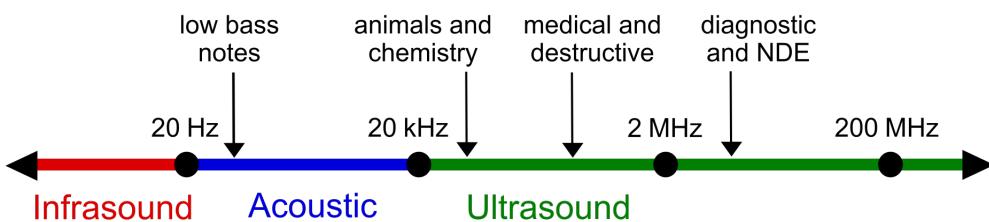


Figure 2.8: Spectrum of sound with annotations of specific regions [72].

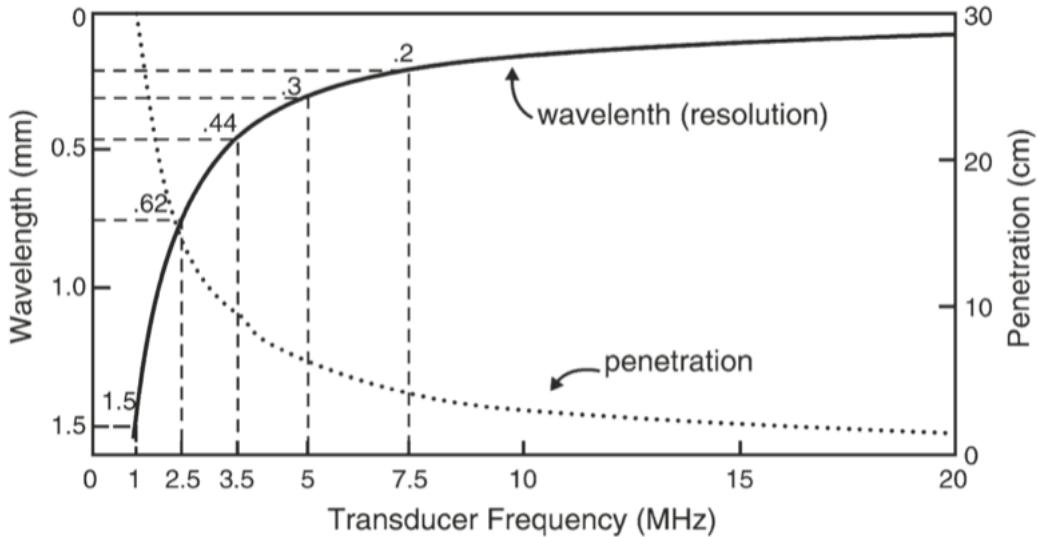


Figure 2.9: Penetration depth and wavelengths for different frequencies [47].

chosen. For the pancreas, on the other hand, 5cm of penetration depth is sufficient and therefore a frequency of 10MHz can be chosen with a better resolution (see Fig. 2.9). Ultrasound interacts with tissues in four different ways: absorption, reflection, refraction, and dispersion. The formation of an ultrasound image is based on reflection, but the other effects influence the image quality as well. The fraction of reflection  $R$  and transmission  $T$  between two interfaces with normal incidence to the interface is defined as

$$R = \left( \frac{Z_2 - Z_1}{Z_2 + Z_1} \right)^2 \quad (2.6)$$

$$T = \frac{4Z_1Z_2}{(Z_1 + Z_2)^2} \quad (2.7)$$

where  $Z_1$  and  $Z_2$  are the acoustic impedance of tissue 1 and 2 respectively, which is defined as

$$Z = \rho c \quad (2.8)$$

where  $\rho$  is the material density and  $c$  the material speed of sound. Hence, the higher the difference of  $Z$  between two tissues, the higher the reflection. The intersection of air and human skin would have a very high reflection fraction. This is also the reason why an ultrasound gel is used. Only a small fraction of the wave would be transmitted into the tissue and the image would be mostly black as the waves cannot travel deep enough into the tissue. Table 2.3 shows examples of the acoustic impedance of human tissues. To create an image, ultrasound devices have so-called *Transducers* which act as a source and detector at the same time. Transducers rely on the piezo-electric effect which states that there are crystals that can convert an electric voltage to mechanical force and vice versa. This allows the transducer to

Tissue	Acoustic Impedance ( $10^6$ )
Air	0.0004
Lung	0.18
Fat	1.34
Liver	1.65
Blood	1.65
Kidney	1.63
Muscle	1.71
Bone	7.8

Table 2.3: Acoustic Impedance values for different tissue types [13].

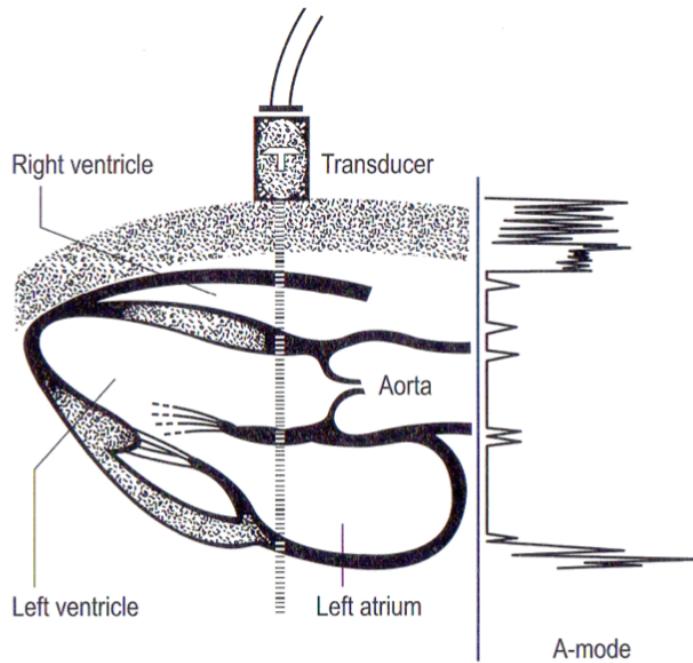
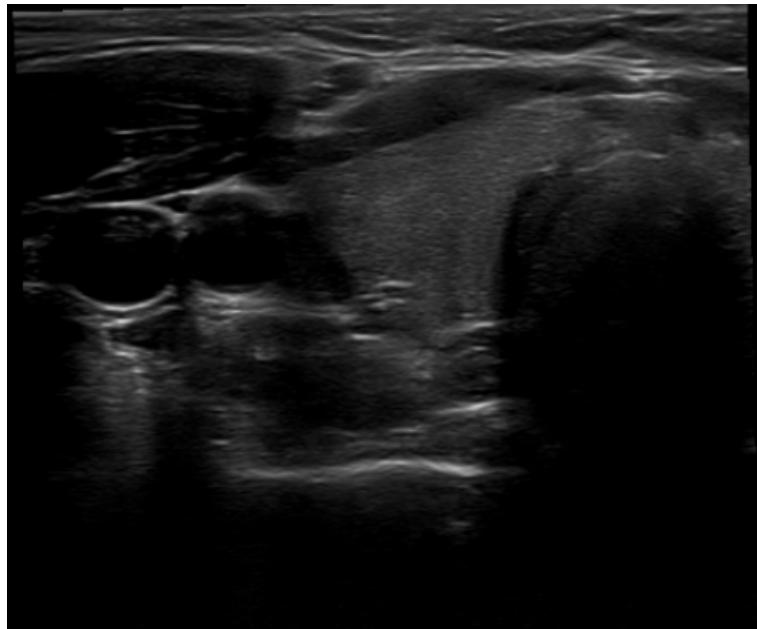
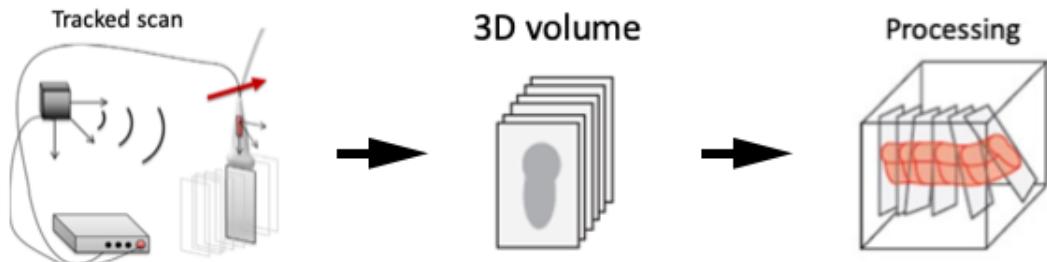


Figure 2.10: Principle of A-mode image generation in ultrasound. [48]

create an ultrasound wave by applying a voltage with desired frequency to convert it into sound waves. The waves are sent into the tissue and immediately after, the transducer *listens* for echos of the sound wave occurring in the tissue at interfaces between types of tissue. As this echo comes back to the transducer, it converts this sound wave back to a voltage that can be measured. In this way, a one-dimensional line called the A-mode of the image can be formed (see Fig. 2.10). To form a two-dimensional image this is repeated for many lines by sweeping the piezo-crystal either mechanically inside of the transducer or electronically by a phased array of piezoelectric elements which focus the ultrasound wave in desired wavefronts or specific points. The final image formed by an ultrasound device can be seen in Fig. 2.11. There is also the possibility to create 3D-US. Matrix 2D arrays can be used to create 3D beamforming natively by having a two-dimensional transducer. This allows for



*Figure 2.11: Ultrasound image of the left side of the neck inside a patient with the thyroid visible.*



*Figure 2.12: Principle how to generate a 3D volume from tracked US images [52]/[35].*

real-time 3D+time data on a small footprint [62]. Another way is a motorized 1D-array imaging transducer that sweeps motorized in the other dimension not yet covered in conventional US. This also allows for real-time 3D-US but suffers from a bigger footprint [26]. The method used in this thesis is the tracked 3D-US, which tracks the transducer freehand. The images together with the tracked information allow for subsequent processing into a 3D volume (see Fig. 2.12). This yields a method where there are no limitations on the field of view and a cheap option for clinics, as they just have to buy this tracking software and hardware as an add-on to the existing US device. The drawbacks are that there is no real-time 3D, as it is processed subsequently [52][35].

### 2.1.4 Computed Tomography

Given the technique of computed tomography (CT) is only used marginally, it will be explained very briefly in this section. Like sound waves, x-rays also follow the Beer-Lambert law, depicted in Eq. 2.5. CT is an imaging modality that displays the attenuation coefficients of pixels tomographically in a volume. This is achieved by an x-ray source that emits x-rays with energies between 90keV and 140keV. These travel through the body and are detected on the other side with an x-ray detector, which works the same way as a gamma camera detector, explained in Section 2.1.2. But instead of PMTs, there are usually charge-couple device (CCD) arrays measuring the scintillation light. The attenuation on one type of tissue can be calculated via Eq. 2.5. To expand it to multiple tissue types, the exponential functions are multiplied infinitesimal small to get the detected intensity  $I_d$ :

$$I_d = I_0 e^{- \int_{\vec{x} \in l} \mu(\vec{x}) ds} \quad (2.9)$$

where  $I_0$  is the intensity before absorption<sup>3</sup>,  $l$  the path travelled and  $\mu$  the attenuation coefficient. This can be rearranged to get the attenuation integral which is the raw detector data in the end:

$$-\int_{\vec{x} \in l} \mu(\vec{x}) ds = \ln \frac{I_0}{I_d} \quad (2.10)$$

The CT device acquires projections over all angles around the body. In theory, there is only the need for the projection angles  $0 - \pi$  of the body because  $\pi - 2\pi$  are equivalent but projecting in the other direction. In reality, however, there are some reconstruction algorithms, that use all of the angles [17]. The main challenge of CT imaging is to reconstruct the 3D distribution of the attenuation coefficient  $\mu$  from only the projection over lines on various angles. There are two mathematical theorems necessary. Dependent on the distance traveled  $\eta$  by the x-rays<sup>4</sup>, the intensity is attenuated differently. After the detected logarithmic intensity at detector element  $\zeta$ , one gets the sum of attenuations  $p_\gamma(\zeta)$  at angle  $\gamma$ . This is called the *Radon Transform*.

$$p_\gamma(\zeta) = \int_0^s \mu(\zeta, \eta) d\eta \quad (2.11)$$

The projections  $p_\gamma(\zeta)$  can be plotted over the angles  $\gamma$ . This is called a sinogram. Additionally, the *Fourier Slice Theorem* is needed for the reconstruction. It states that one-dimensional Fourier transformation  $P_\gamma(q)$  of the measured projection  $p_\gamma(\zeta)$

<sup>3</sup>known from calibration measurements

<sup>4</sup> $\eta = 0$  at source and  $\eta = s$  at detector

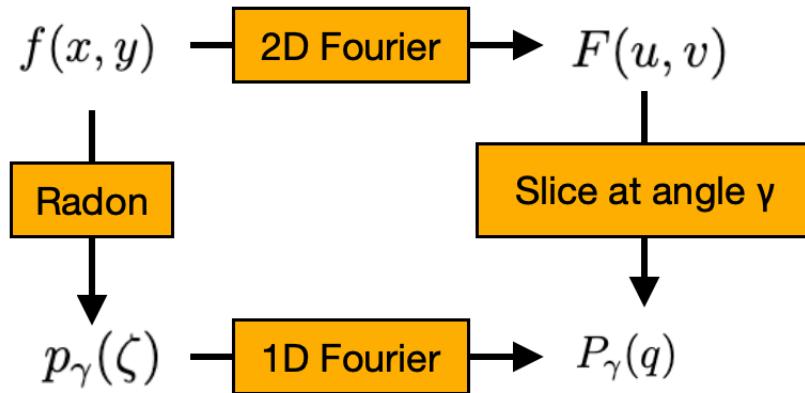


Figure 2.13: Illustration of the Fourier slice theorem in reconstruction.

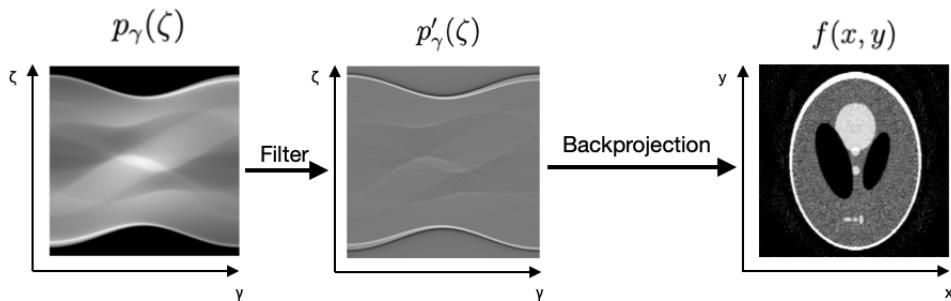


Figure 2.14: Steps involved in the filtered back projection algorithm.

is equivalent to a line in the two-dimensional Fourier transformation  $F(u, v)$  of the wanted image  $f(x, y)$  under angle  $\gamma$ , where  $u = q\cos(\gamma)$  and  $v = q\sin(\gamma)$ . In theory, one could fill the Fourier space  $F(u, v)$  with all angles  $\gamma$  and do an inverse Fourier transform to get the image  $f(x, y)$ , which is the distribution of  $\mu$  in this slice. All of that is illustrated in Fig. 2.13. This can be repeated with all detector lines to get a 3D volume. In practice, however, this is not done because of interpolation problems. Instead, the *Filtered Backprojection* (FBP) is used. It has two underlying key ideas: Firstly, one can mathematically bypass the interpolation in the Fourier domain by summing over the inverse Fourier transforms of the individual projections. Secondly, a filter in the Fourier domain is necessary to compensate for the amplification of low frequencies by the back projection. In conclusion, the steps of the FBP are calculating the sinogram out of the logarithmic detector line intensities, Fourier transform the sinogram, apply a filter in the Fourier domain, inverse Fourier-transform this sinogram, and back-project the filtered sinogram. All these steps with images can be seen in Fig. 2.14. Finally, because the attenuation coefficients  $\mu$  are displayed in gray scales and every detector might be slightly different, the dimensionless *Hounsfield*

*Unit* (HU) is introduced and used:

$$HU = \frac{\mu - \mu_{water}}{\mu_{water} - \mu_{air}} 1000 \quad (2.12)$$

This way, air is at  $HU = -1000$  and water at  $HU = 0$ . [36] [60]

## 2.2 GATE

Monte-Carlo simulations rely on repeated random sampling to get a result [61] and have been used for over 50 years when a closed mathematical solution is not possible or the simulation gets computationally too expensive [20]. An example is the estimation of the irrational number  $\pi$ . Consider a circle with radius  $\frac{r}{2}$  inside of a square with a side of length  $r$ . Knowing the formulas of the area of a circle and a square, one could derive that  $\pi = 4 \frac{A_{circle}}{A_{square}}$ . Now we can randomly generate points inside of the square, and count the number  $N$  of points in the complete square and  $M$ , the number of points being inside the circle. Hence, we can estimate  $\pi$  by  $\langle \pi \rangle = 4 \frac{M}{N}$ . The more points we generate, the more accurate the result will be. This concept can be adapted to more complicated processes such as estimating integrals or simulating the passage of particles through matter. For this purpose, a well-validated object-oriented C++ toolkit was developed called GEANT4 (GEometryANdTracking) [1]. However, this toolkit is hard to use, and tailoring it to a specific problem if one has no prior experience with C++ or this toolkit in general, is very complex. Hence, a user-friendly Monte-Carlo toolkit to accommodate complex scanner geometries was developed by the OpenGATE collaboration starting in 2004 called GATE (Geant4 Application for Tomographic Emission) [31]. To use GATE, no prior knowledge of C++ or Geant4 is needed, as it uses its own descriptive macro language. It is also possible to perform time-dependent phenomena such as source or detector movement. To validate the implemented model in GATE, one needs to replicate the same experimental setup in the real world and compare it to the simulation result. The GATE code is available open-source, is still under development, and updates are released frequently. For example, it was extended to use for modeling of CT and radiotherapy [32], and python implementations [59]. There is also a mailing list available where GATE users can discuss problems and possible solutions. Additionally, docker images and plug-and-play virtual machine configurations are available. All of this is making it very attractive to use for Monte-Carlo simulations in medical research.

## 2.3 Deep Learning

Due to the explosion of data and advances in the development of graphical processing units (GPUs), the research field of deep learning has risen tremendously. Just recently, ChatGPT [10, 46], a tool powered by deep learning, gained broad public attention. Deep learning is a subfield of machine learning where functions are modeled by a so-called neural network. Certain elements in deep learning are vaguely inspired by neural circuits in biology. Supervised learning is a type of machine learning that involves providing a neural network with known pairs of inputs and outputs [38]. The goal is to train the network so that it can produce the desired output (also called "ground truth") when presented with an unknown input. The network is structured with several layers, including an input layer, hidden layers, and an output layer. During training, the network is shown the input, produces an output, and then the output is compared to the desired output. This comparison is evaluated with a scalar error (loss), which the model tries to reduce by modifying its internal adjustable parameters, called weights. This weight adjustment is done using the back-propagation algorithm, which calculates the gradient vector of the loss with respect to the weights [56]. This process is called the backward pass. A simple version of a feedforward neural network can be seen in Fig. 2.15. Often, networks consist of more than a million parameters and have many different architectures.

### 2.3.1 Convolutional Neural Networks

One building block that is used mostly for networks that take images as input are convolutional filters. They were introduced as the vanilla fully connected (FC) neural network would explode in size with images as input. For example, if you take an image with  $256 \times 256$  resolution, the input layer would have 65536 values, one for each pixel. These are then multiplied in the next layer with the weights of each unit. So each unit has 65536 weights. If the layer has a size of 65536 units, this would yield more than four billion parameters, just with one layer. Of course, one could reduce the size of the hidden layers, but this comes with the trade-off of a lower expression power of the network.

To reduce the number of weights, convolutions are introduced. Moreover, they provide the right inductive bias to exploit the spatially local patterns of image data making learning more sample-efficient. A convolution is a mathematical operation on two functions, sliding one function over the other. It is defined as:

$$(f * g)(t) := \int_{-\infty}^{\infty} f(\tau)g(t - \tau)d\tau \quad (2.13)$$

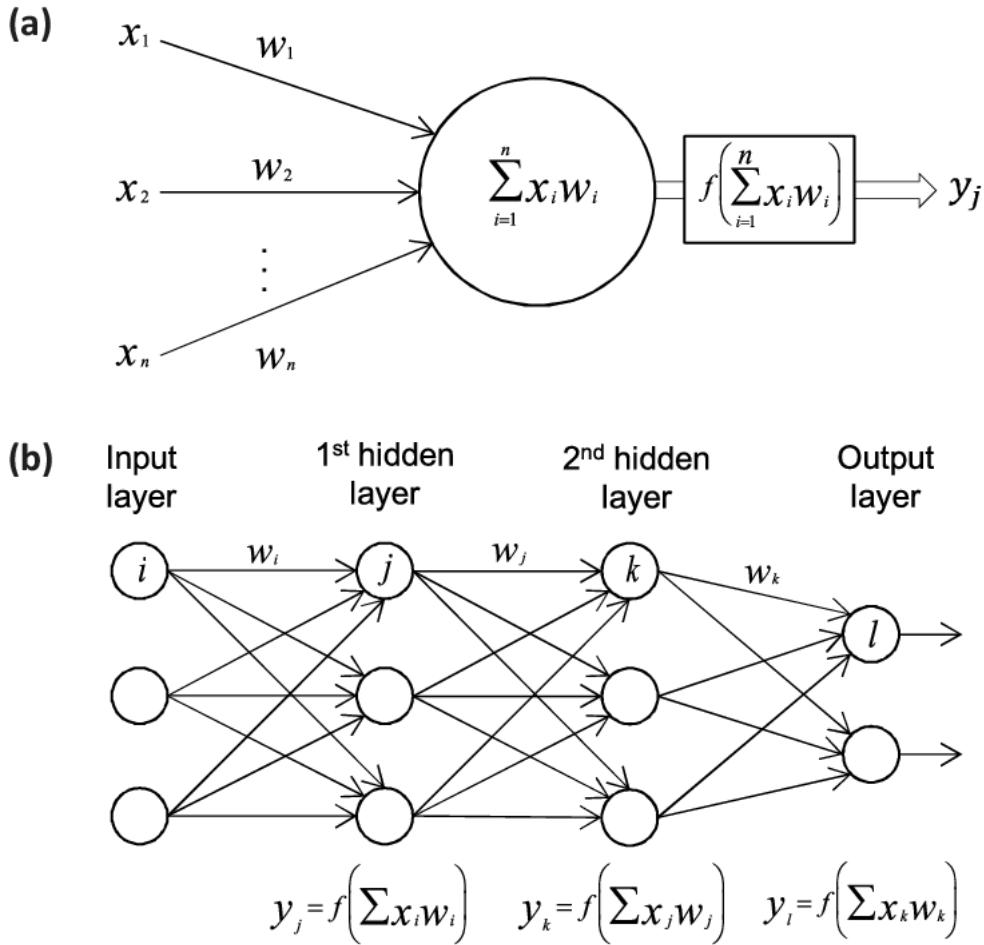


Figure 2.15: The figure illustrates the constituent elements of a neural network in (a). A unit is depicted that accepts inputs  $x$  and aggregates them by means of weighted summation utilizing the weights  $w$ . Following this, an activation function  $f$  is employed to introduce non-linearities, thereby generating the output  $y$ , which subsequently serves as the input for the next layer (b). The hyperparameters of the neural network, such as the number of layers, the type of activation function, and the interconnections, impact the network's efficacy significantly. This figure has been sourced from [69].

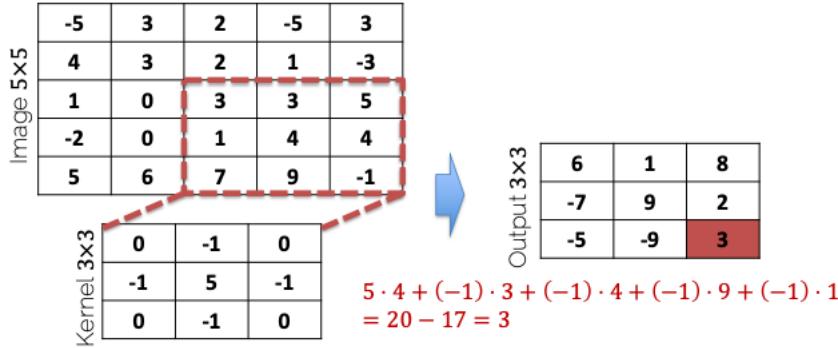


Figure 2.16: Example of a 2D convolution using a  $3 \times 3$  filter on a  $5 \times 5$  image as input. The output is a  $3 \times 3$  image. In this operation, only 9 weights are necessary, as the kernel has size  $3 \times 3$  [45].

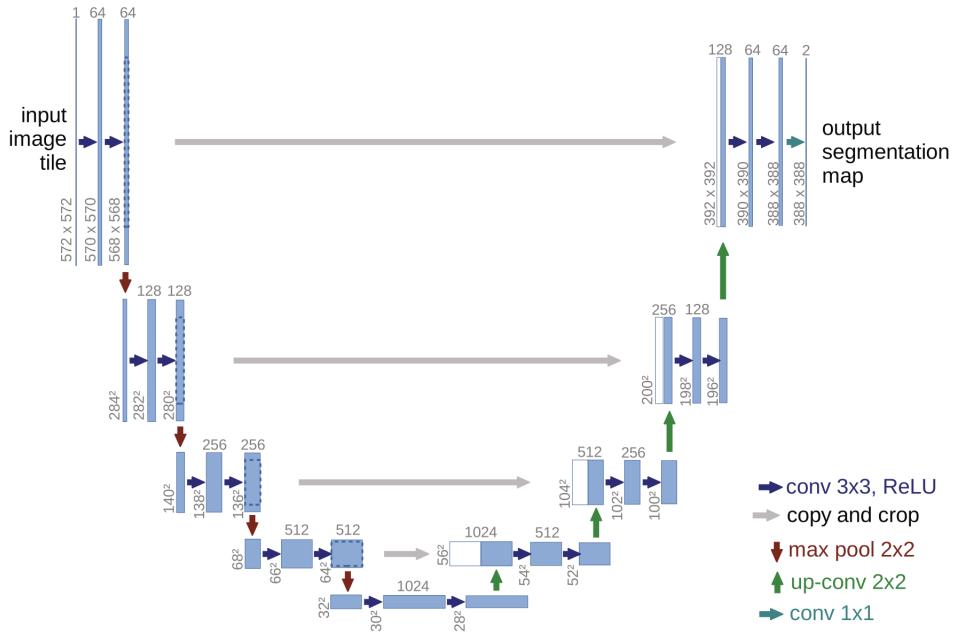
In the context of deep learning the data is often discrete. Hence, the discrete convolution is a kernel, also called a filter, which is slid over the input and multiplied with each parameter of the kernel. These products are summed up and used as the output. A 2D example is depicted in Fig. 2.16. This can be extended to multiple dimensions. Usually, the output is called a feature map. Again, this output is summed with a bias term and used as an input of an activation function of choice to introduce non-linearity. Important factors that influence the output are stride and padding. Stride is the stepsize of the kernel. Padding means putting zeros around the input image to keep the dimensions of the input and output the same as convolutions reduce the spatial extent depending on the filter size. Convolutional layers are also often used in combination with a max pooling layer where the maximum value instead of the sum of products within the kernels' receptive field is selected.

## U-net

The U-net is a very frequently used CNN architecture, mainly used in image segmentation [55]. It consists of a contracting path to capture context and a symmetric expanding path that enables localization (see Fig. 2.17). The contracting path consists of repeated application of two  $3 \times 3$  convolutions which are unpadded, followed by a rectified linear unit (ReLU) activation function which is defined as:

$$\text{ReLU}(z) = \max(0, z) \quad (2.14)$$

This is followed by a  $2 \times 2$  max pooling operation with stride 2 to downsample. At each step in the contracting path, the number of channels doubles and the spatial resolution decreases. The expansion path is the symmetric opposite of the contract-



*Figure 2.17: U-net architecture: blue boxes mean a multi-channel feature map. The number of channels is denoted on top of each box and the x-y-size is at the lower left edge. White boxes represent copied feature maps.*

ing path, consisting of an upsampling of the feature channels followed by  $2 \times 2$  up convolutions that halve the number of channels every step, and two  $3 \times 3$  convolutions followed by a ReLU layer. Additionally, at every step, the corresponding cropped<sup>5</sup> feature map of the contracting path is concatenated. As the U-net was initially designed for image segmentation, the last layer is a layer of  $1 \times 1$  convolutions that maps each of the 64-component feature vector to the desired number of classes to segment.

### 2.3.2 Conditional Diffusion Models

Recently, many generative models were brought to the public's attention through image generation models like DALLE-2 [51] or Stable Diffusion [54]. They can generate new images via a diffusion process that can be conditioned, for example, with a text prompt or another image, such as a sketch. Diffusion models generate samples from an empirical data distribution by gradual denoising of a sample from a standard gaussian noise distribution [29, 63]. This means that once trained, they can generate infinite amounts of new images. In general, the diffusion process does not need any conditions, and can also generate new samples without inputs. The process is split up into two sub-processes: the forward and the reverse diffusion

<sup>5</sup>Due to loss of border pixels in every convolution.

process. In the forward process, the input is a data point sampled from a real data distribution  $\mathbf{x}_0 \sim q(\mathbf{x})$ . A small amount of Gaussian noise is added to the sample in  $T$  steps, yielding a sequence of noisy samples  $\mathbf{x}_1, \dots, \mathbf{x}_T$ , where the step size is controlled by a variance schedule  $\{\beta_t \in (0, 1)\}_{t=1}^T$ . At  $T \rightarrow \infty$ ,  $\mathbf{x}_T$  is equivalent to isotropic Gaussian noise. Using some simplifications and the reparameterization trick  $\mathbf{x}_t$  can be sampled at any time step  $t$  in a closed form:

$$q(\mathbf{x}_t | \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t} \mathbf{x}_0, (1 - \bar{\alpha}_t) \mathbb{I}) \quad (2.15)$$

where  $\bar{\alpha}_t = \prod_{i=1}^t (1 - \beta_i)$ . If the forward diffusion process can be reversed and sampled from  $q(\mathbf{x}_{t-1} | \mathbf{x}_t)$ , a true sample can be recreated from a Gaussian noise input  $\mathbf{x}_T$ . Because  $q(\mathbf{x}_{t-1} | \mathbf{x}_t)$  needs the entire dataset to be easily estimated, a model  $p_\theta$  needs to be learned to approximate the conditional probabilities. The variational lower bound is used to optimize the negative log-likelihood yielding to the simplified training objective:

$$L_{DM} = \mathbb{E}_{x, \epsilon \sim \mathcal{N}(0, 1), t} [\|\epsilon - \epsilon_\theta(x_t, t)\|_2^2] \quad (2.16)$$

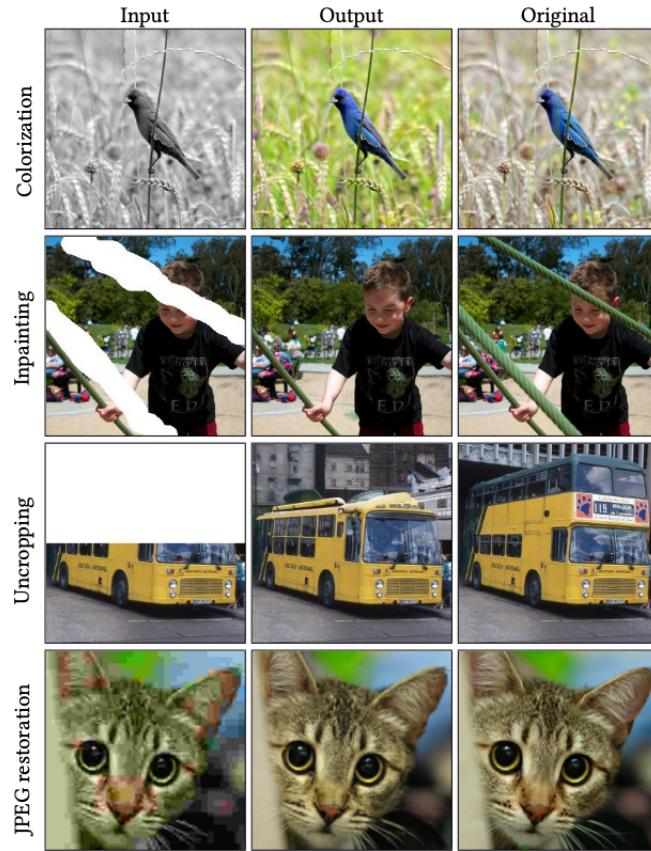
where  $\epsilon_\theta(x_t, t); t = 1 \dots T$ , is an equally weighted sequence of denoising autoencoders trained to predict a denoised variant of their input  $x_t$ , which is a noisy version of the input  $x$ , and  $t$  uniformly sampled from  $\{1, \dots, T\}$ . In the following, two network architectures using the diffusion process together with conditions to generate images, are presented.

## Palette

Palette was introduced in 2022 by Google Research and is a unified framework for image-to-image translation tasks such as colorization, inpainting, uncropping, or JPEG restoration, based on conditional diffusion models [57]. Their image-to-image conditional diffusion model has the form  $p(y | x)$ , where  $x$  and  $y$  are images. Given a training output image  $y$ , the model generates a noisy version  $\tilde{y}$  and a neural network  $f_\theta$  is trained to denoise  $\tilde{y}$  given  $x$  a noise level indicator  $\gamma$ , yielding the loss similar to Eq. 2.16:

$$\mathbb{E}_{x, y} \mathbb{E}_{\epsilon \sim \mathcal{N}(0, 1)} \mathbb{E}_\gamma \|f_\theta(x, \underbrace{\sqrt{\gamma} y + \sqrt{1 - \gamma} \epsilon}_{\tilde{y}}, \gamma) - \epsilon\|_2^2 \quad (2.17)$$

As an architecture for the neural network, they use a U-Net architecture as described in Section 2.3.1 with additional conditioning of the source image via concatenation [58]. Results of their model can be seen in Fig. 2.18. Additionally, they analyzed the impact of L1 and L2 loss in training and self-attention in diffusion models. They



*Figure 2.18: Results of Palette: Image-to-Image translation tasks with examples of colorization, inpainting, uncropping, and JPEG restoration [57].*

found that models trained with L2 loss have greater sample diversity than those trained with the L1 loss. The self-attention analysis yielded that global self-attention offers better performance than fully-convolutional alternatives. Local self-attention performs worse than fully-convolutional alternatives.

### Latent Diffusion Models

Latent diffusion models (LDM) were introduced in 2021 and are used for image synthesis on various conditions when limited computational resources are available [54]. To lower the computational demand they utilize an autoencoding model which learns a space that is perceptually equivalent to the image space but with a significantly reduced computational complexity. The computationally intense diffusion process is then performed in this latent space. In the end, the sample is decoded back into image space. This autoencoder is trained with a combination of a perceptual loss and a patch-based adversarial objective to ensure the reconstructions are confined to the image manifold by enforcing local realism. Given an image,  $x \in \mathbb{R}^{H \times W \times 3}$  in RGB space, an encoder  $\mathcal{E}$  encodes  $x$  into the latent representation  $z = \mathcal{E}(x)$ , and a decoder  $\mathcal{D}$  reconstructs it back into the image space. The encoder downsamples

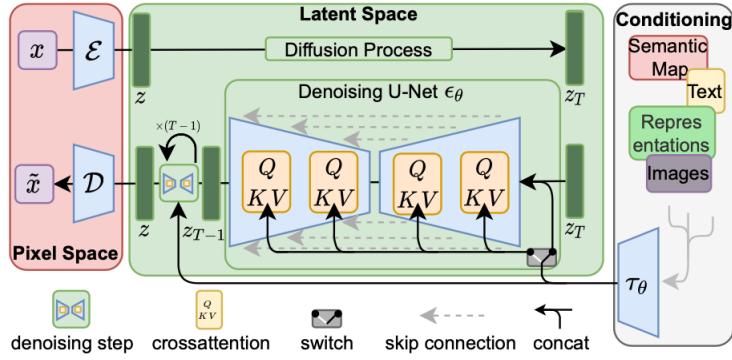


Figure 2.19: Architecture of the latent diffusion model with conditioning via concatenation or a general cross-attention mechanism [54].

the image by a factor  $f = H/h = W/w$ , where the different downsampling factors  $f$  were investigated. To conclude, the reconstruction is  $\tilde{x} = \mathcal{D}(z) = \mathcal{D}(\mathcal{E}(x))$ , where  $z \in \mathbb{R}^{h \times w \times c}$ . Additionally, a KL-regularization and a VQ-regularization were investigated. In latent space, the objective is

$$L_{LDM} = \mathbb{E}_{x, \epsilon \sim \mathcal{N}(0,1), t} [\|\epsilon - \epsilon_\theta(z_t, t)\|_2^2] \quad (2.18)$$

As in Palette, the neural networks  $\epsilon_\theta$  architecture of choice is a time conditional U-net. The conditioning is realized by augmenting the underlying U-net backbone with the cross-attention mechanism, which works for learning attention-based models with various input modalities. The input  $y$  is pre-processed with a domain-specific encoder  $\tau_\theta$  that puts  $y$  in an intermediate representation  $\tau_\theta(y) \in \mathbb{R}^{M \times d_r}$ . This is then mapped to the intermediate layers of the U-net via a cross-attention layer. This then yields the conditional LDM loss

$$L_{LDM} = \mathbb{E}_{\mathcal{E}(x), y, \epsilon \sim \mathcal{N}(0,1), t} [\|\epsilon - \epsilon_\theta(z_t, t, \tau_\theta(y))\|_2^2] \quad (2.19)$$

where  $\tau_\theta$  and  $\epsilon_\theta$  are jointly optimized. The architecture is visualized in Fig. 2.19. They found that the LDMs trained with VQ-regularized latent spaces achieve better sampling quality compared to KL-regularization and a downsampling factor  $f$  between 4 and 16 yielded the best tradeoff between efficiency and perceptually faithful results.

# Chapter 3

## Methods

### 3.1 Approach

This thesis aims to get a 3D voxelized activity map (pseudo-SPECT) of a specific patient when only given a segmented 3D-US<sup>1</sup>, the relative pose of the 3D-US to the camera, and a scintigraphy. To achieve this, several steps need to be taken (see Fig. 3.1).

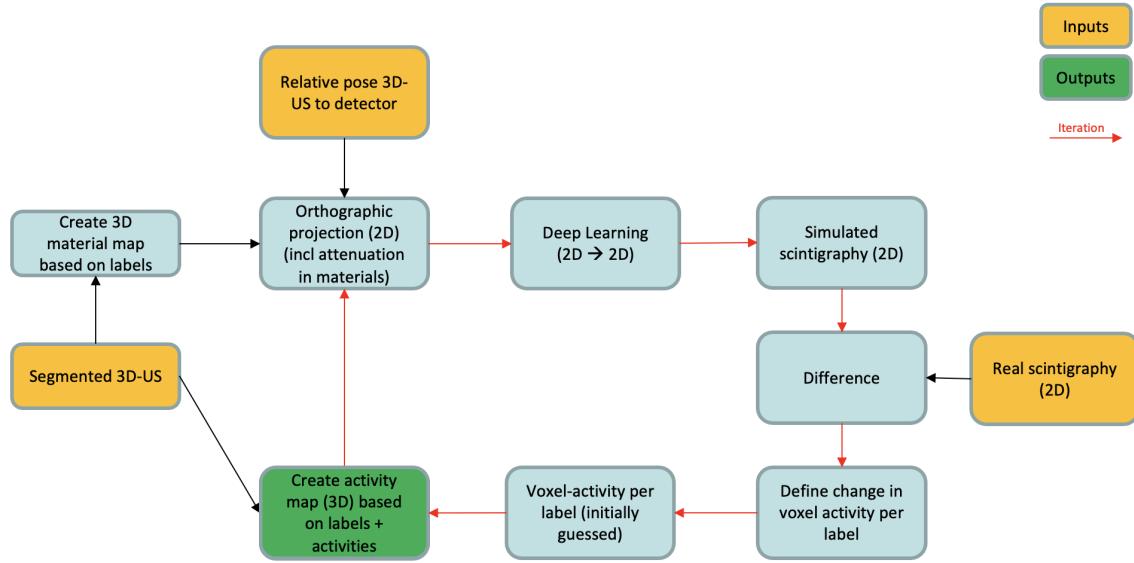
First, the 3D ultrasound volume of the thyroid and (if applicable) additional nodules inside is segmented, which is not within the scope of this thesis and is assumed to be already completed. The 3D segmentation mask is required to convert the 3D-US into a 3D material map, which can be used directly as input for GATE or as a foundation to create a synthetic CT. To generate the synthetic CT from the labels, each label is assigned Hounsfield Unit ranges corresponding to its respective materials, with additional noise for realism. The number of labels determines the level of realism in the synthetic CT. As our primary focus was on a plastic phantom, we were able to skip this step and directly acquire a real CT, as explained in detail in Section 4.1.3.

Next, the relative pose of the 3D-US to the camera needs to be determined, which is achieved using an optical tracking system attached to the camera and the phantom, as elaborated in Section 4.1.2. The optical tracking system yields the position of the phantom relative to the camera.

Subsequently, an initial 3D activity map has to be created based on the labels from the 3D segmentation mask, which includes the thyroid and four nodules in our case. We initially estimate some activity for the labels, which may not accurately reflect the true output activity initially but serves as an initialization that can be iteratively

---

<sup>1</sup>MRI or CT can also be used. It just has to be a 3D image of the target anatomical region, namely the neck.



*Figure 3.1: Overview of the proposed algorithm to estimate a 3D voxelized activity map: Step 1: A 3D material map and an initially guessed activity map are created based on a segmented 3D-US. Step 2: The 3D activity map, the material map, and the relative pose of the US to the camera are used to create an orthographic projection. Step 3: The orthographic projection is the input for a style transfer neural network, yielding a simulated scintigraphy. Step 4: The simulated scintigraphy is compared to the real scintigraphy. Step 5: Based on the difference, the change per label is calculated. Step 6: This change is added to the initially guessed 3D activity map. Step 7: Repeat Step 2 - Step 6 until convergence.*

refined. Since the synthetic CT and the 3D activity map are based on the same segmentation, they are inherently registered. In summary, we now have a 3D activity map, a synthetic CT, and the relative position and pose of the phantom/patient to the camera. These three inputs are utilized to generate a 2D orthographic projection (coarse simulated scintigraphy) that employs the activity map as a gamma ray source and the synthetic CT as an attenuation material. This is accomplished using a CPU-based rendering script, as explained in Section 3.4.

As the orthographic projection is subject to certain assumptions, it is not truly representative. Hence, the projection is used as input for a deep neural network with a style transfer approach to enhance its realism. The training of the neural network is examined in more detail in Section 3.5. The output of the neural network is a further improved realistic simulated scintigraphy based on the assumed activity. As the assumed activity is manually initialized and may not be accurate, the simulated scintigraphy will differ from the real and given scintigraphy. The difference is utilized to calculate a change in the activity for each label, which is then added to our initially assumed activity for each label in the 3D activity map.

The above-described process is now repeated iteratively with the utilization of the updated 3D activity map, which is expected to be more accurate than the previous

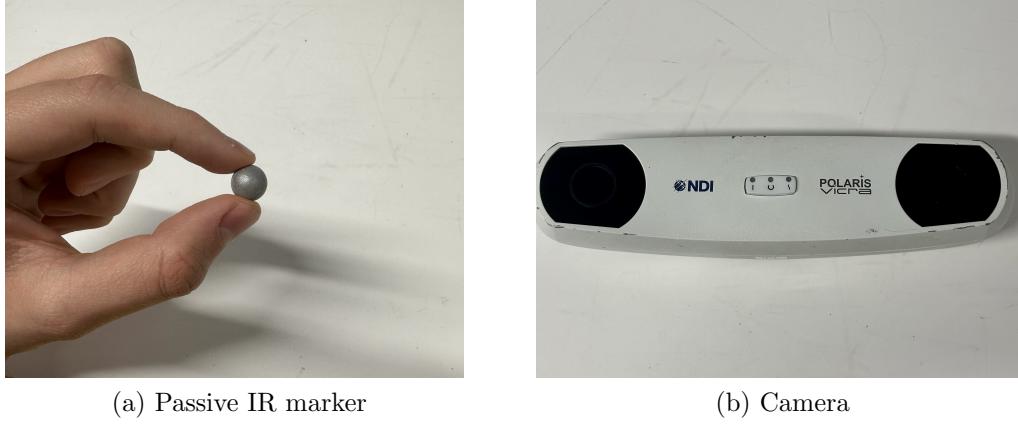
version, as it is based on the newly adapted 3D activity map. This iterative loop is repeated until convergence, which is defined as the point where the difference between the real and simulated scintigraphy falls below a predefined threshold (see Section 3.6). Ultimately, this process is expected to yield a single 3D activity map that generates a scintigraphy which is consistent with the real 3D activity distribution. However, it is important to note that this problem is inherently ill-posed, meaning there are infinitely many possible solutions to the reconstruction task due to the limited information available from a single projection. To address the ill-posed nature of the problem, we rely on additional constraints from the 3D-US to narrow down the range of solutions, for example, by constraining the activity to segmented areas. Unfortunately, due to the limited availability of scintigraphy data for training purposes, the neural network utilized in this thesis was unable to be trained using a sufficient amount of real scintigraphies. Hence, the initial phase of this thesis focused on generating a realistic Monte Carlo simulator using the GATE software to generate simulated scintigraphies, which serve as the ground truth for training the neural network. Once the neural network is trained, the Monte Carlo simulator is no longer necessary. The simulator takes input in the form of a material map, a radioactive source distribution, and a camera model to generate realistic simulated scintigraphies. The details of the implementation of this simulator are thoroughly explained in Section 3.3. In order to validate the accuracy of the simulations, real-world measurements were obtained and compared against the simulated data, as elaborated in Section 4.1.

While it is theoretically possible to carry out the entire project using only the GATE simulator without using the neural network, the computational time required for GATE simulations is prohibitively long for clinical application. Depending on the level of activity, a single scintigraphy simulation using GATE can take several days to weeks to complete, which is impractical considering the need for repeated simulations at each iteration step.

## 3.2 Optical Tracking

Optical tracking is used to measure the location of active or passive markers in three dimensions via infrared (IR) light. Active markers transmit their own infrared light whereas passive markers are highly reflective spheres that require an external IR source. This IR light is detected by at least two charged couple device (CCD) cameras, where each camera captures a 2D image of the scene. The relative position of the cameras to each other is fixed per construction and therefore known. In this

thesis, the Optical tracking system *Polaris Vicra* is used with passive markers. In Fig. 3.2 passive markers and the two cameras including the IR sensor can be seen. These two cameras both capture two coordinates of the marker in the 2D image.



(a) Passive IR marker

(b) Camera

*Figure 3.2: Left: Highly reflective passive infrared (IR) marker. Right: The dual camera + IR sources*

From those two images a 3D position has to be triangulated. The cameras are assumed to be calibrated. The software provided by Polaris does everything automatically, and no calculations are needed. Nevertheless, the basic principle will be explained in the following. There are a few different ways to solve this triangulation problem, such as the mid-point method, via the essential matrix, or with a direct linear transformation. The latter will now be explained in more detail.

First, we switch to homogeneous coordinates that are often used in computer vision, as in this representation, points at infinity can be represented in finite coordinates. Homogeneous coordinates require one additional component in the vector. The triangulation problem basically states that given a set of noisy matched points  $\{\vec{U}_1 = [u_1, v_1, 1], \vec{U}_2 = [u_2, v_2, 1]\}$  and camera matrices  $P_1$  and  $P_2$ , the 3D point  $\vec{X} = [x, y, z, w]$  can be estimated. The camera matrices  $P_1$  and  $P_2$  can be obtained by camera calibration, which can be done with multiple techniques. Usually, it is done by acquiring multiple images from different perspectives of a known pattern, e.g., a checkerboard.  $\vec{U}_1$  can be written as

$$\vec{U}_1 = \alpha P_1 \vec{X} \quad (3.1)$$

where  $\alpha$  is the scale factor. Since  $\vec{U}_1$  and  $P_1\vec{X}$  are parallel vectors, the cross product should be zero, yielding

$$\begin{bmatrix} u_1 \\ v_1 \\ 1 \end{bmatrix} \times \begin{bmatrix} \vec{p}_1\vec{X} \\ \vec{p}_2\vec{X} \\ \vec{p}_3\vec{X} \end{bmatrix} = \begin{bmatrix} v_1\vec{p}_3\vec{X} - \vec{p}_2\vec{X} \\ \vec{p}_1\vec{X} - u_1\vec{p}_3\vec{X} \\ u_1\vec{p}_2\vec{X} - v_1\vec{p}_1\vec{X} \end{bmatrix} = \begin{bmatrix} v_1\vec{p}_3 - \vec{p}_2 \\ \vec{p}_1 - u_1\vec{p}_3 \\ u_1\vec{p}_2 - v_1\vec{p}_1 \end{bmatrix} \vec{X} = A\vec{X} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (3.2)$$

where  $p_i$  is the four dimensional vector of row i in matrix  $P_1$ . The third row of matrix A is a linear combination of the first and second row <sup>2</sup>. Hence, the third row can be dropped. This yields two equations for three unknowns, so at this point, it is not solvable, which is expected since we have only considered coordinates from a single camera yet. Since we have two cameras, we can extend matrix A to have more rows analog to the first two rows. In fact, one can add as many camera views as wanted to matrix A. To summarize, we now have the equation

$$A\vec{X} = \begin{bmatrix} v_1\vec{p}_3 - \vec{p}_2 \\ \vec{p}_1 - u_1\vec{p}_3 \\ v_2\vec{p}'_3 - \vec{p}'_2 \\ \vec{p}'_1 - u_2\vec{p}'_3 \end{bmatrix} \vec{X} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (3.3)$$

where  $p'_i$  is the four dimensional vector of row i in matrix  $P_2$ . Matrix A has dimensions  $2n \times 4$  where n is the number of camera views included. As the measurements are noisy, this equation cannot be solved. However, the closest solution can be solved using the *total least squares* method. It tries to minimize  $\|A\vec{X}\|^2$  with the constraint  $\|\vec{X}\|^2 = 1$ . The solution to this problem uses singular value decomposition yielding the eigenvector corresponding to the smallest eigenvalue of  $A^\top A$ . [27].

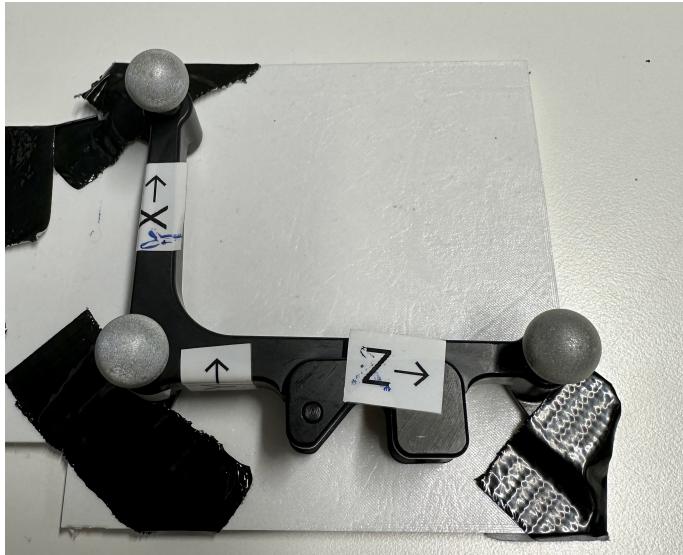
From this, we can get the 3D position from the two corresponding points in the images from one marker. However, the position of one marker is not sufficient, as it only yields the position, but not the orientation. Because of that, a group of markers can be registered as a combined target and then tracked. In this case, one can also calculate the orientation of the target. One target used in this thesis can be seen in Fig. 3.3.

### 3.3 Ground Truth Generation with GATE

To generate ground truth scintigraphies, GATE version 9.2 was used. It is possible to visualize geometries using the implemented user interface, as seen in Fig. 3.4.

---

<sup>2</sup> $u_1$  times the first row plus  $v_1$  times the second row.



*Figure 3.3: A group of markers combined into a target to get the position and orientation of the target. The coordinate system has to be defined when registering the target.*

The complete simulation script requires the specification of the following eight components.

### 1. Camera geometry

The camera model is defined using the GATE macro language. Several geometries are available, together with a predefined list of materials that can be used and extended with custom materials. A camera model geometry is usually made out of a crystal, a collimator, covers of the camera, PMTs (back-compartment), and shielding.

**Crystal:** The crystal is usually a rectangular block of scintillating materials. Frequently used materials are NaI(Tl), BGO, LSO, and GSO.

**Collimator:** Many different shapes of collimators are available. In this thesis, parallel hexagonal holes were assumed, as this is the standard for collimators produced via micro casting [22]. To generate the hexagonal pattern in GATE, a solid block of lead is created. Afterward, the hexagonal holes made out of air are inserted in this block via cubic arrays with the parameters  $P_y$  and  $P_z$ . They are calculated via

$$P_y = D + ST \quad (3.4)$$

$$P_z = \sqrt{3} (D + ST) \quad (3.5)$$

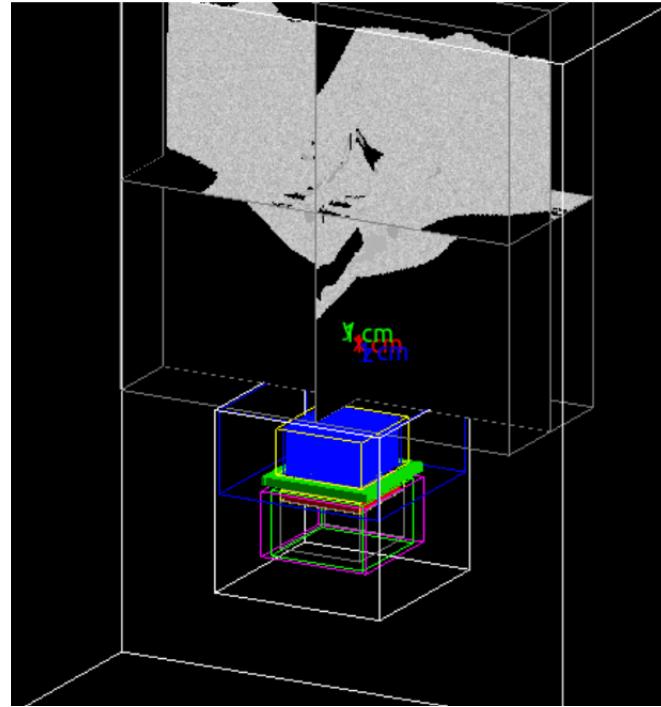


Figure 3.4: Example of a camera and a CT-phantom implemented and visualized in GATE.

where  $D$  is the hole size and  $ST$  is the septa thickness. First, one hexagonal hole of air with a given radius and height is generated. Then, a `cubicArray` repeater is used for a grid of parallel hexagonal holes. To cover the whole collimator, this needs to be repeated in the x-direction and in the z-direction. For the repeat vector,  $P_y$  and  $P_z$  from Eq. 3.4 and Eq. 3.5 can be used as  $x$  and  $z$  values respectively, and one for the  $y$  value. To fill in the leftover holes, a `linear` repeat vector with repeat number 2 and a repeat vector of  $(\frac{P_z}{2}, 0, \frac{P_y}{2})$  is used. This procedure is illustrated in Fig. 3.5. The whole camera with aluminum cover, crystal, and collimator can be seen in Fig. 3.6.

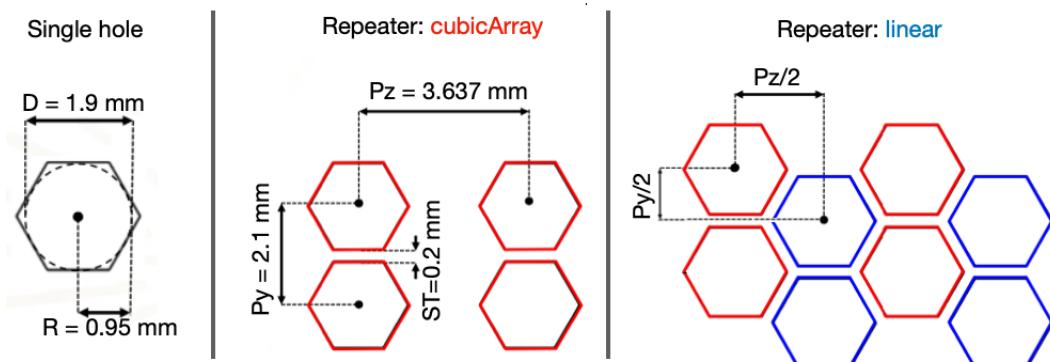
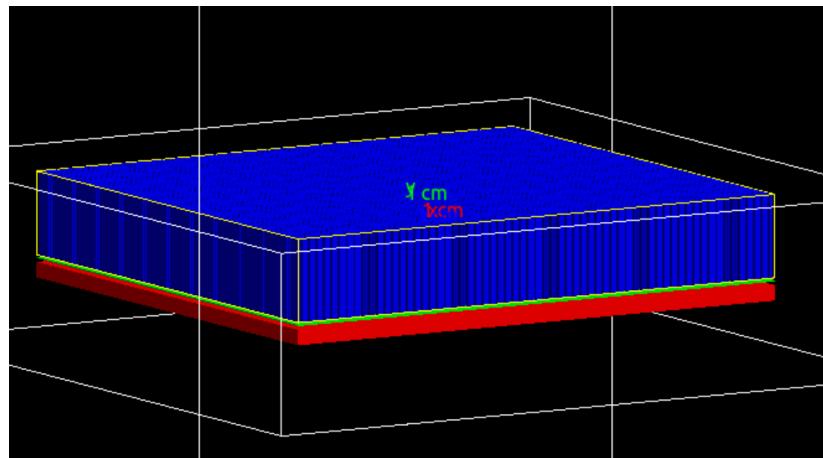
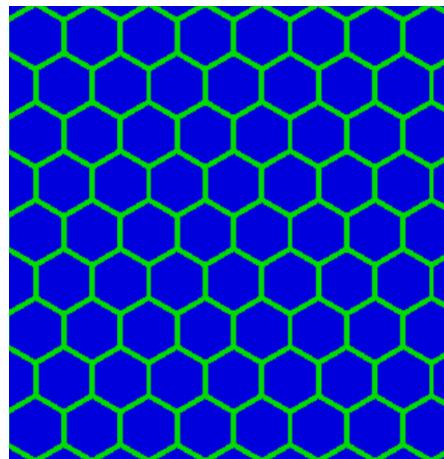


Figure 3.5: A hole of air and two repeaters are used to generate a hexagonal parallel hole collimator. The figure is adapted from [33].



(a) Whole camera geometry in GATE.



(b) Top-down view on collimator.

*Figure 3.6: The whole camera as a 3D model in GATE with the crystal in red, the cover in green, and the collimator holes in blue. The lead block is drawn as a wireframe to see inside.*

## 2. Phantom geometry

To generate the phantom geometry, there is the possibility to either build them with GATE commands or with imported files. In this thesis, Meta Header Files (MHD) were used and imported via `ImageNestedParameterisedVolume` into GATE because of its high efficiency. Once, they are inside GATE, one could either use a table that converts the imported CT directly to specific materials or create a custom materials conversions table.

## 3. Digitizer (detector response)

The digitizer is made out of several modules to convert the hit inside the crystal into the pixel value of the camera. This means it mimics the readout with the PMTs and digitalization. The *adder* module sums the hits generated in the crystal. Next, the *readout* module is used to sum the energy deposited in the crystal and determines the position of the crystal with the highest energy deposited. Also, a *spatial blurring* module is used to define the spatial resolution. It assumes a Gaussian distribution defined by its width  $\sigma$ . Additionally, the *pileup* module and *deadtime* can be used. Energy blurring and quantum efficiency are introduced via the *crystal blurring* module. Finally, an energy window can be set according to the camera.

## 4. Time parameters (start and stop times) and 5. Output format

GATE simulations can either be limited by time with a start and stop time, or via a maximum amount of generated particles. Several output formats, such as ASCII and ROOT, or projections can be used.

## 6. Physics processes

The physics can be defined either manually with every physical process wanted to be listed, or pre-made physics lists can be used.

## 7. Radioactive source(s)

Similar to the camera geometries, sources can be generated with geometries inside of GATE. For the geometry of the source, external files can be imported additionally. In both cases, the source's emission type, angle, and energy need to be specified.

## 8. Verbosity

Verbosity commands can be used to log and print several parameters of the simulation.

## 3.4 Rendering

The rendering aims to generate an orthographic projection of the 3D source activity distribution including material attenuation. To achieve this, a raycasting algorithm is used. The idea is that for every pixel in the image, one ray is cast through a volume and samples the volume in predefined steps. Finally, the sampled values are converted into pixel values via a so-called shader. To calculate the ray vectors, the camera needs to be defined via an extrinsic and intrinsic matrix. The extrinsic matrix defines the camera pose and consists of the rotation and translation of the camera. It is constructed as

$$\hat{\mathbf{C}}_{ext} = \begin{pmatrix} \hat{\mathbf{R}} & \vec{t} \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (3.6)$$

where  $\hat{\mathbf{R}}$  is the rotation matrix and  $\vec{t}$  the translation vector.

In this project, the optical tracking result<sup>3</sup> is used to get these parameters. For the intrinsic matrix, the camera model needs to be specified. For an optimal pinhole camera, it is defined as

$$\hat{\mathbf{C}}_{int} = \begin{pmatrix} f_x & s & x_0 \\ 0 & f_y & y_0 \\ 0 & 0 & 1 \end{pmatrix} \quad (3.7)$$

where  $f_x$  and  $f_y$  are the focal length parameters for x and y direction,  $s$  the skew, and  $[x_0, y_0]$  the center of the image (see Fig. 3.7). Next, the rays that originate from the camera's position and are going through each pixel in the image plane, are calculated with the camera matrices. A bounding box of the target volume is defined and entry points and exit points are calculated for each ray. The ray is sampled inside of the volume and a customized shader is used to calculate the pixel values using the label values as the source intensity and a CT for attenuation (see Fig. 3.8). Implementations are explained in Section 5.2.

As this orthographic projection suffers from many assumptions, such as no scattering

---

<sup>3</sup>Or GATE parameters in ground truth generation.

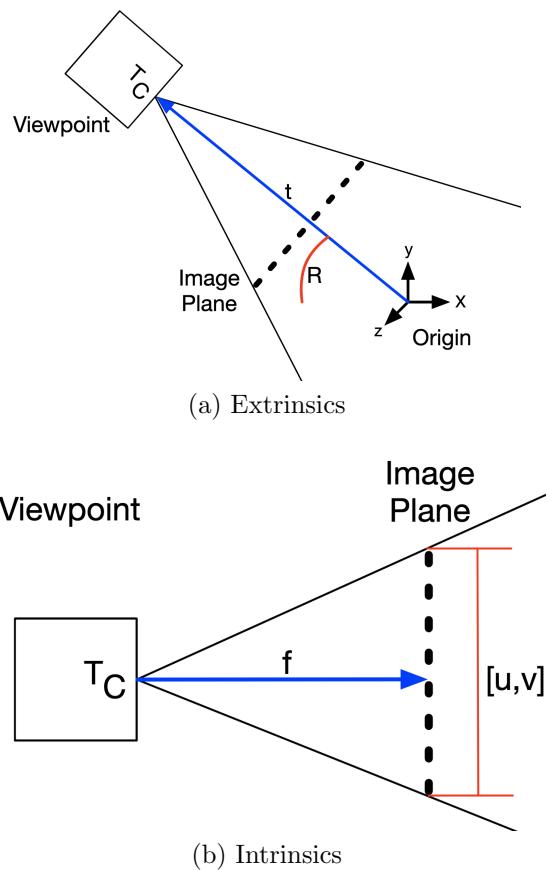


Figure 3.7: Both extrinsic (a) and intrinsic (b) parameters are visualized [19].

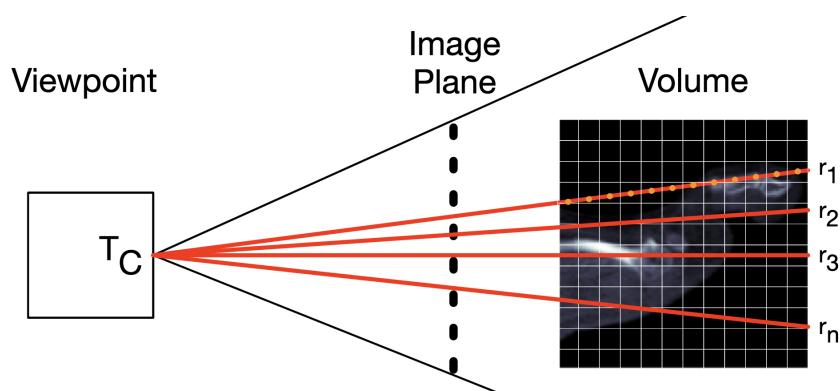


Figure 3.8: Sketch of the raycasting principle. Rays are depicted in red with  $r_n$  and the camera is  $T_C$  [19].

and no collimator, the result is not realistic, but still sufficient. The deep learning framework ideally makes this image more realistic and mimics the processes not included in the rendering.

## 3.5 Deep Learning

The neural network’s task is to generate a more realistic scintigraphy, given the output of the rendering algorithm. To train the network we have generated a dataset of matching pairs of renderings and GATE simulations (see Section 4.2). The GATE simulations act as the ground truth. The whole dataset was normalized between 0 and 1 to improve performance. Additionally, data augmentation was used during training to improve the model’s generalization capabilities despite having a relatively small dataset of 945 samples. To fasten the training and avoid overfitting, an early-stopping callback was implemented. The architecture of choice was a pre-implemented U-net (see Section 2.3.1) from the *MONAI* [12] package trained with the *Pytorch Lightning* framework [21] with a supervised learning approach. To find the best hyperparameters, *Weights & Biases* [9] sweeps were used. To train the final model, a random training, validation, and test split of 80%, 10%, and 10% was chosen.

## 3.6 Iteration

The iteration loop aims to change the activity of the voxels for each label to match two images together. One of those images is the reference image, which in our case is the reference GATE simulation. The other image is the current output of the neural network.

To update the activity in each voxel, a novel algorithm was developed:

1. Project each label of the 3D label map onto the same imaging plane as GATE or the scintigraphy.
2. Calculate one mask for each label with this projection.
3. Multiply masks with current and reference image separately.
4. Sum pixel values of images of step 3 and store them in two lists.
5. Calculate the difference between two lists, multiply them by a factor and divide them by the number of voxels the label has in the 3D label map.

Implementation details and the algorithm are explained in more detail in Section 5.4.

# Chapter 4

## Experimental Setup

### 4.1 Measurement

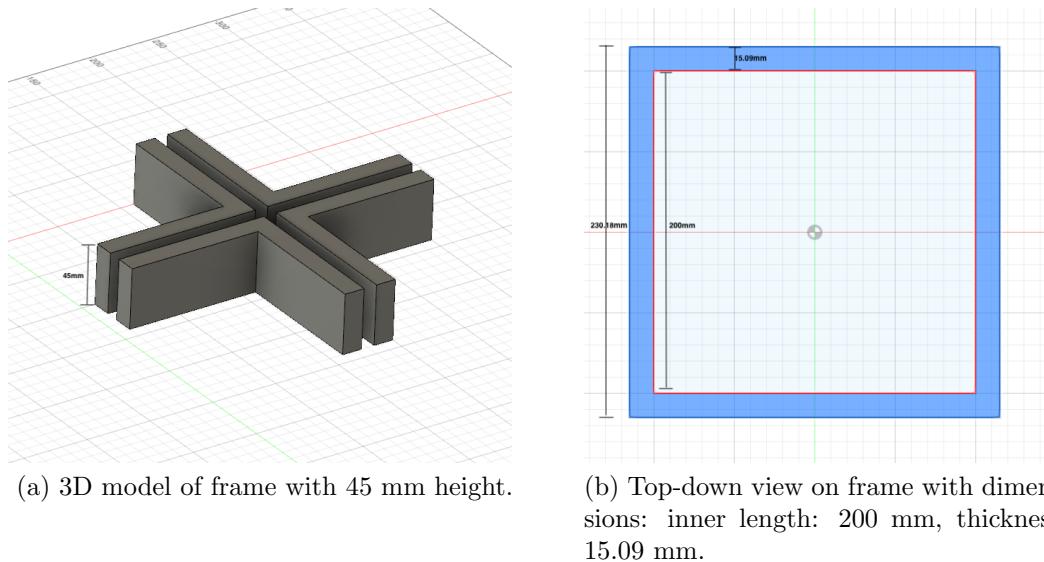
To validate that the GATE simulations do in fact match with reality, real measurements are required for comparison. To realistically simulate uptakes in the thyroid, a thyroid phantom with four nodules was 3D-printed. Additionally, syringes were filled with different amounts of  $^{99m}\text{Tc}$ -pertechnetate. In all cases, this provides the ground truth of activity distribution, as it can be filled as desired. To get the ground truth about the position of the activity with respect to the camera, optical tracking was used. To achieve that, additional mounts for the tracking targets had to be designed and 3D printed. A CT of the phantom was acquired to import the geometry of the phantom into the simulation and to use it for the labeling of the activity distribution. Additionally, CTs of the phantom were used to determine the quality of the 3D printing. Lastly, the measurements were taken on the *Mediso Nucline TH22* planar gamma camera.

#### 4.1.1 3D Printing

To build 3D structures fast and with little cost, 3D printing is a good solution. For all of the 3D printed structures in the thesis, the 3D printer *Creatlity CR-10S* was used. A frame for the camera, a mounting plate for the phantom, and the phantom itself had to be 3D printed.

##### Frame

The frame is the mount of the tracking target on the gamma camera. As the collimator casing sticks out of the camera, a squared frame is easily attachable. On the frame, the tracking target can be attached in one corner. The frame was

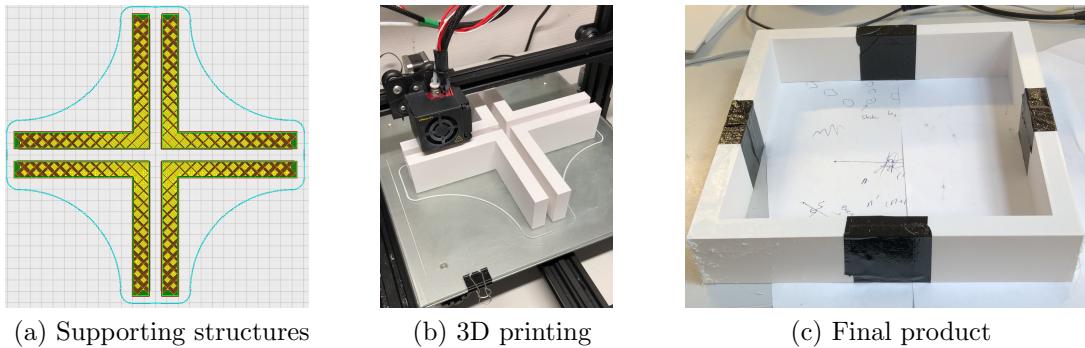


*Figure 4.1: Sketch and dimensions of the frame model.*

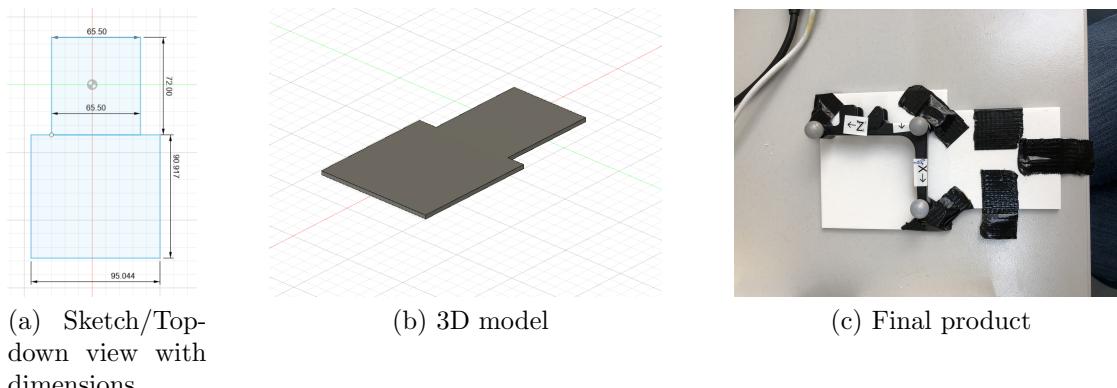
designed using *AutoDesk Fusion 360* with an inner side length of 200 mm. The thickness of the frame can be chosen arbitrarily, as it is just for stability. We have visually chosen a thickness of 15.09 mm, yielding an outer length of the frame of 230.18 mm. The height was chosen to be 45 mm (see Fig. 4.1). If the 3D printer is not perfectly leveled/calibrated, the print has a level offset, which is most prominent at points far away from the center, the corners. This yields in either printing in the air or with too much pressure on the bed, which breaks the 3D printing. With the available 3D printer leveling was only possible to a certain degree. Due to the frame structure having a big spatial expansion, it was chosen to split the frame into four equal parts and attached together afterward. This way the highest quality of the 3d print was achieved. The frame was chosen not to be filled solid, but with a grid as a supporting structure to speed up the printing and save material. The 3D printing took almost 24 hours. The supporting structure inside the frame, the 3D printing and the final product can be seen in Fig. 4.2.

## Plates

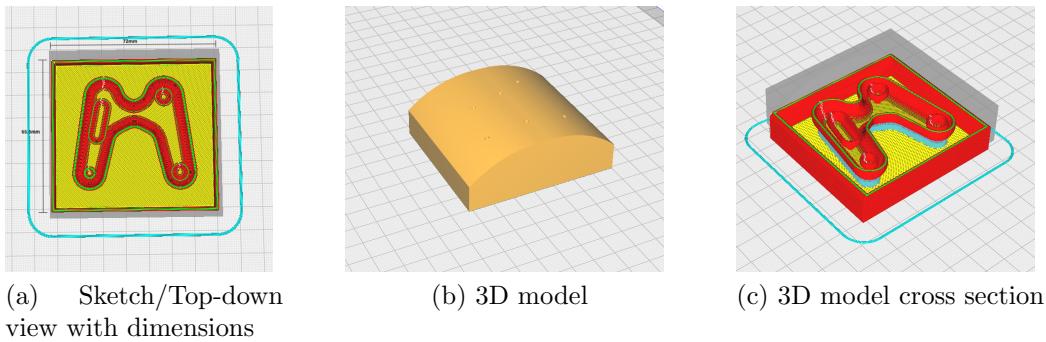
The plates are used to mount the tracking targets to the frame on the camera and the phantom. For the mounting plate of the tracking target on the phantom, it had to fit the dimensions of the phantom. The other part of the plate, where the tracking target is attached, can be of any dimension, as long as the target fits on it, and does not obstruct views of the tracking camera. The thickness was chosen to be 3 mm, as this was sufficient for structural stability. The sketches with dimensions, the 3D model and the printed product are depicted in Fig. 4.3. The mounting plate for the



*Figure 4.2: The process of 3D printing. Figure (a) shows the supporting structures as the infill, (b) shows the printing itself and (c) depicts the final product without the tracking target attached.*



*Figure 4.3: The process of 3D printing of the mounts. Figure (a) shows the sketches with dimensions, (b) shows 3D model of the plate and (c) depicts the final product with the tracking target attached but without the phantom, which would be on the other side.*



*Figure 4.4: The 3D models of the phantom. Figure (a) shows the top-down view with dimensions, (b) shows 3D model of the phantom and (c) shows a cross section with the 4 nodules visible inside the thyroid.*

frame was not custom-designed but leftovers from previous 3D printing iterations of the phantom plate were used. There is no custom shape that needs to be fitted as long as the tracking target is attached to the frame leveled. This can be seen together with all other 3D printed parts in Fig. 4.6.

## Phantom

The thyroid phantom was designed in a realistic size and shape and includes four nodules. The dimensions of the whole structure are 72 mm × 65.5 mm × 26 mm. Each nodule and the thyroid itself have a hole cutout to insert liquids with a syringe. As the holes are quite small, it is easy to get liquids in with a syringe, whereas retrieval of the liquids is not possible. Hence, this is a disposable product and for each measurement, a new phantom has to be printed. It took many iterations until sufficient quality and a decent time to 3D print were reached. In the end, PETG was used as a material, 230°C printing temperature, 75°C build plate temperature and a 20% infill density. The supporting structures below the thyroid were necessary for leakproof printing in the bottom layers. The “roof” of the structure also has some overhangs, but printing quality was sufficient. Also, a high-quality print for the roof was not necessary as errors can be seen visually and discarded. The models and printed results can be seen in Fig. 4.4 and Fig. 4.5 respectively.

### 4.1.2 Tracking

To realistically simulate a scintigraphy, the position of the phantom and source with respect to the camera are necessary. Once known, it can be digitally constructed inside GATE or used as an input for a deep neural network. Optical Tracking (OT) was implemented to get the phantom’s position during the measurements. The

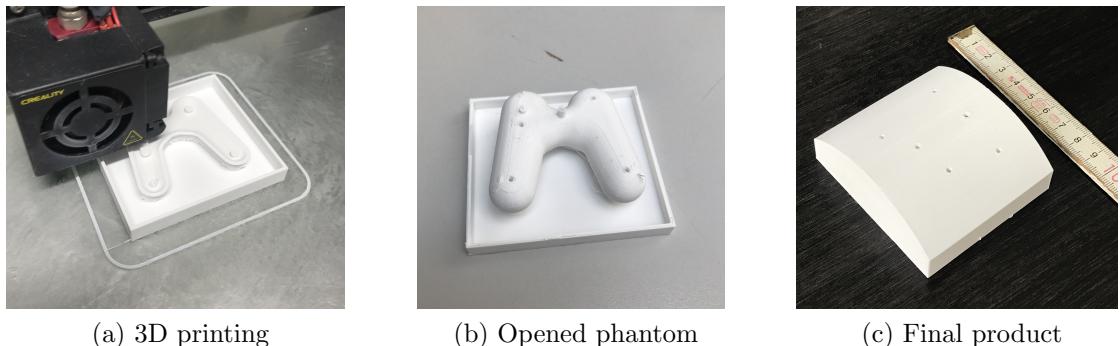
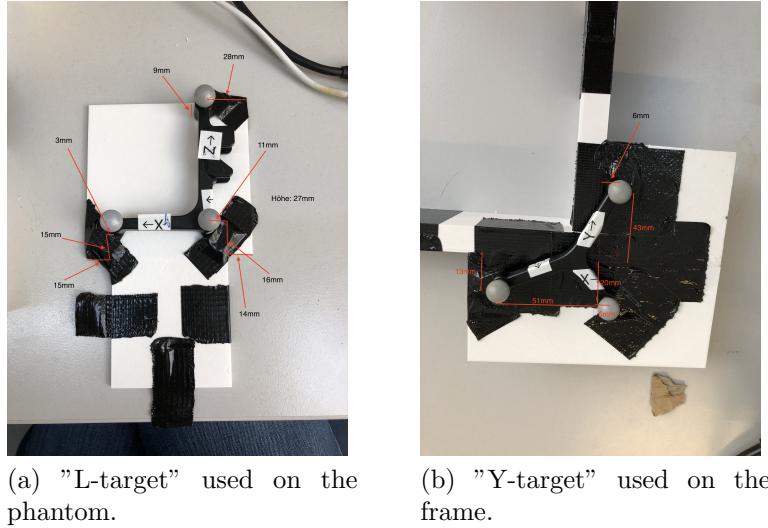


Figure 4.5: The 3D printing of the phantom. Figure (a) shows the 3D printing with the supporting structure slightly visible, (b) shows the open up-ed phantom and (c) shows the final product printed with a ruler for reference. Also the holes are visible to inject liquids.



Figure 4.6: All 3D printed parts including the frame with the tracking target attached, the mounting plate for the phantom with tracking target attached, and two phantoms.



*Figure 4.7: The dimensions of the targets and their positions are used in the transformations.*

camera *Polaris Vicra* and its included software was used for tracking. In total, two tracking targets were used with one attached to the camera and the other to the phantom. The dimensions and positions of the target on the frame and the target on the phantom plate can be seen in Fig. 4.7. Both were registered with the software *NDI 6D Architect*. This is necessary for the tracking software to know which IR-reflective spheres belong to which target and to define a coordinate system for each target. In the tracking software, one can define one tool as the global reference so the position of the other tool is always relative to the reference tool. This is realized by the multiplication of the  ${}^{cam}T_{T_2}$ , referring to the transformation from the camera to target 2, with the inverse of  ${}^{cam}T_{T_1}$  which is the transformation from the camera to target 1.

$${}^1T_{T_2} = {}^{cam}T_{T_1}^{-1} \cdot {}^{cam}T_{T_2} \quad (4.1)$$

We have set the target on the frame as the global reference frame. Hence, the tracking software yields the position of the phantom's target coordinate system origin relative to the target on the frame coordinate systems origin. For the simulations, we need to know the position of the centre of the phantom relative to the surface of the camera. As we know the position of the targets on the camera and the phantom, this information can be calculated in the same manner as seen in Eq. 4.1. To get an intuition for these kinds of transformations, graphs can be used. The graph of the tracking system can be seen in Fig. 4.8. For visualization purposes, the transformations are abbreviated.  $\hat{T}_1$  refers to the transformation from the middle point of the surface of the 3D printed frame to the target of the frame. This relation

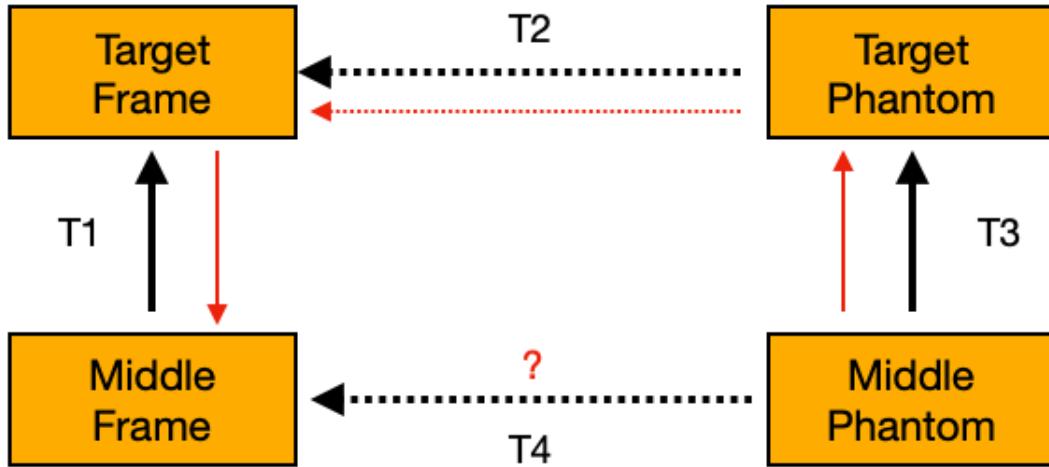


Figure 4.8: The graph visualizes the tracking with its transformations. Dashed arrows refer to dynamic relations whereas solid arrows mean this transformation is fixed and does not change. To get the unknown transformation, one can walk on the trace of known transformations and concatenate them on the way. This is visualized with the red arrows.

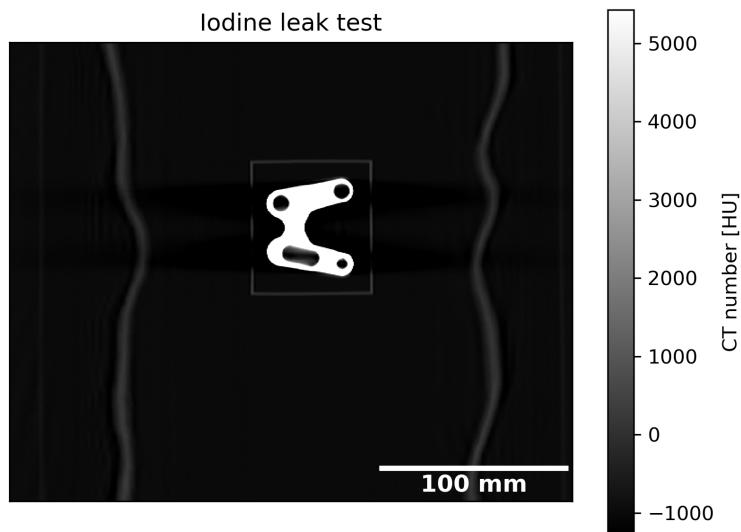
is fixed, as the target is fixated on the frame. The same goes for  $\hat{T}_3$ , which is the transformation from the middle point of the bottom of the phantom to the target of the phantom. The last known transformation  $\hat{T}_2$  is the dynamic transformation from the target of the phantom to the target of the frame. This is obtained by the tracking software. To get the transformation  $\hat{T}_4$ , which is from the middle of the phantom to the middle of the frame, the transformations need to be applied as shown in Eq. 4.2.

$$\hat{T}_4 = \hat{T}_1^{-1} \cdot \hat{T}_2 \cdot \hat{T}_3 \quad (4.2)$$

The transformations  $\hat{T}_1$  and  $\hat{T}_3$  can be calculated with the dimensions depicted in Fig. 4.7 and yield

$$\hat{T}_1 = \begin{pmatrix} 1 & 0 & 0 & -121 \\ 0 & 1 & 0 & 135 \\ 0 & 0 & 1 & -27 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \hat{T}_3 = \begin{pmatrix} 1 & 0 & 0 & -52 \\ 0 & 1 & 0 & 21.5 \\ 0 & 0 & 1 & -30 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (4.3)$$

Using Eq. 4.2,  $\hat{T}_4$  can be calculated. This procedure was validated with several measurements with known positions in 3D. We are dealing with an error of up to 4mm, because everything is measured by hand and the tracking system has an intrinsic error.



*Figure 4.9: Coronal slice of a CT of the 3D-printed phantom filled with iodine for higher contrast. This was done to test if there are any leaks in the 3D-printing of the phantom. We did not find any leaks. The lines left and right of the phantom is the bed of the CT device.*

### 4.1.3 CT of Phantom

The 3D-printed phantom was scanned with a *Siemens Healthineers SOMATOM go.Top* CT for three reasons:

1. Used in the simulations to simulate attenuation.
2. Used as a ground truth for labeling the thyroid and nodules. It is used as the activity distribution later
3. Checking for leaks visible in the CT to evaluate the 3D printing quality. For that, we filled the phantom with iodine

For the CT, the following parameters were chosen: a voltage of 70 kVp, a current of 225 mA, a slice thickness of 0.8 mm and a Br64f reconstruction filter. For the iodine leak test, a higher current of 318 mA was chosen. In the CT for labeling and the attenuation map, multiple phantoms were scanned at once and post-processed<sup>1</sup> afterwards. Both the CT of the iodine leak test and the final CT used for labeling and attenuation can be seen in Fig. 4.9 and Fig. 4.10 respectively.

---

<sup>1</sup>cropped and rotated, and sometimes rescaled to different voxel dimensions.

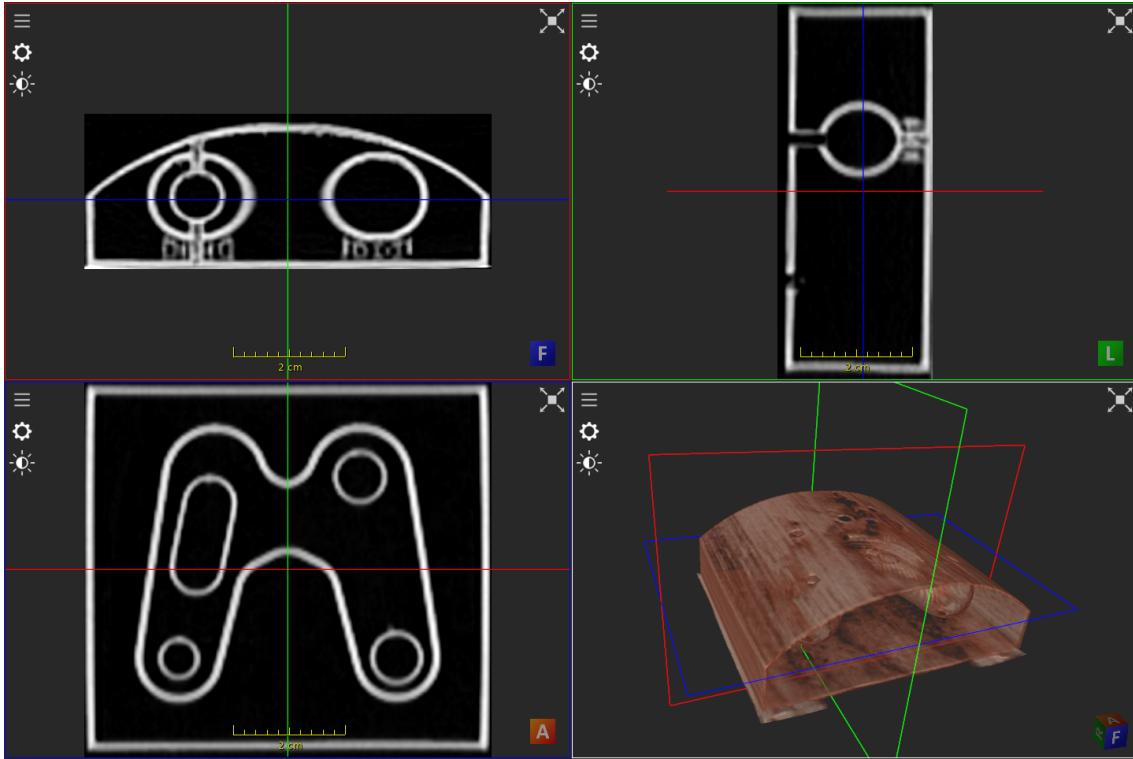
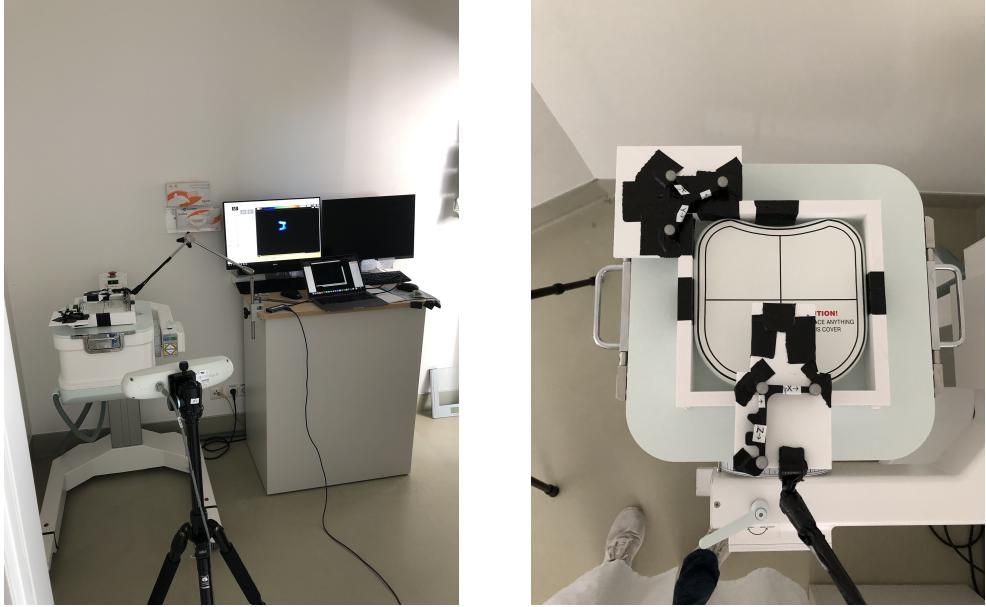


Figure 4.10: The final CT of the phantom used for attenuation calculations and labelling. Coronal, axial and sagittal planes are visualized together with a 3D rendering.

#### 4.1.4 Scintigraphy of Phantom

For the scintigraphies of the 3D-printed phantom, the tracking targets, their 3D-printed mounts, and the camera frame were used. A metallic gimbal arm was used to hold the phantom. The camera for tracking was placed in front of the *Mediso Nucline TH22* planar gamma camera, so the tracking targets were in the field of view. The position of the tracking camera is not important as only the position of the phantom with respect to the static gamma camera frame is measured. This measurement does not change if the tracking camera is moved, as long as all tracking targets are in the line of sight. The setup can be seen in Fig. 4.11. Before the phantom was attached to the tracking target and the gimbal arm, it was filled with  $^{99m}\text{Tc}$ -pertechnetate by an MTRA (medical-technical radiology assistant). To measure the activity injected, a syringe was filled with  $^{99m}\text{Tc}$ -pertechnetate, which was then put inside the *ISOMED 2010* activimeter. Then, the phantom was filled with the measured  $^{99m}\text{Tc}$ -pertechnetate, and filled up with a saline solution (NaCl). To prevent the phantom from leaking, glue pads were used to seal the insertion holes. Afterward, the empty  $^{99m}\text{Tc}$ -pertechnetate syringe is measured again with the activimeter to measure the leftover activity in the syringe. That way, the total activity injected into the phantom can be calculated by simply subtracting the leftover ac-



(a) Gamma camera together with the optical tracking system. The laptop is used for the optical tracking and the workstation for the scintigraphy.

(b) Topdown view of gamma camera with frame attached to it and the phantom attached to the gimbal arm.

*Figure 4.11: The whole setup of the phantom measurements on the gamma camera.*

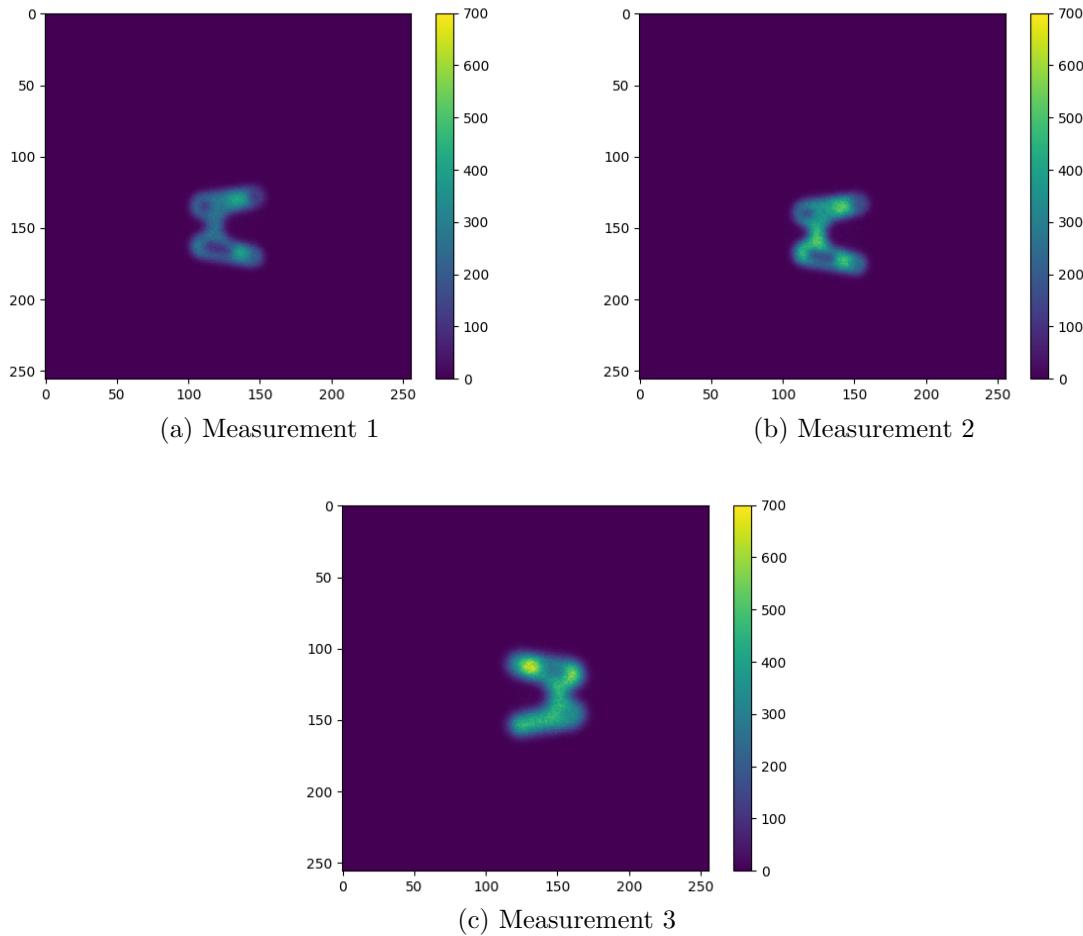
tivity from the initial activity. As the insertion holes for the phantom are small, the activity can be injected easily with a syringe, but can not easily be extracted afterward. Hence, the phantoms were used only once, and thrown away afterward. For every measurement, a new 3D-printed phantom is used. For the scintigraphy, the standard *Tc-99m Schilddrüse* protocol of the manufacturer was used. As no patients are involved, no *Leerspritze*, *Para*, and *Marker* measurements were taken. The *Schilddrüse* measurement measures for 300 s with the energy window of 126.45 keV - 154.55 keV.

Three measurements of the phantom were taken with three different activities and positions. In Table 4.1, the activity and the optical tracking results of the measurements are summarized. The final scintigraphies can be seen in Fig. 4.12. It

Measurement	Injected activity [MBq]	$T_x$ [mm]	$T_y$ [mm]	$T_z$ [mm]	$R_x$	$R_y$	$R_z$
1	14.374	-104.22	204.98	26.26	1.6	3.0	92.1
2	23.083	-99.68	208.71	28.31	0.4	3.0	91.8
3	27.587	-94.27	188.84	78.81	-1.7	1.3	91.4

*Table 4.1: Parameters of the three phantom measurements.  $T_x$ ,  $T_y$ , and  $T_z$  are the translation and  $R_x$ ,  $R_y$ , and  $R_z$  the Euler rotation angles from the tracking software.*

should be noted, that on measurement 3 the phantom accidentally rotated by 180 degrees on the tracking target. However, this was accounted for in the simulations



*Figure 4.12:* The three measurements taken with the gamma camera. The pixel values represent the number of hits detected. Parameters for each measurement are summarized in table 4.1.

by applying a 180-degree rotation to it.

#### 4.1.5 Scintigraphy of Syringe

Additionally, scintigraphies of only syringes filled with  $^{99m}\text{Tc}$ -pertechnetate and NaCl were acquired. As this was done only for hit rate parameter tuning, no tracking was performed. A standard 1ml syringe with a diameter of 5mm and a length of 2.1cm was used. The experimental setup can be seen in Fig. 4.13. The measurements taken are depicted in Fig. 4.14.

## 4.2 Dataset for Deep Learning

The dataset consists of 945 samples of matching GATE-Rendering pairs. They were generated using *Amazon Web Services (AWS)* and a local cluster. To do this, a



Figure 4.13: 1ml syringe used for scintigraphies.

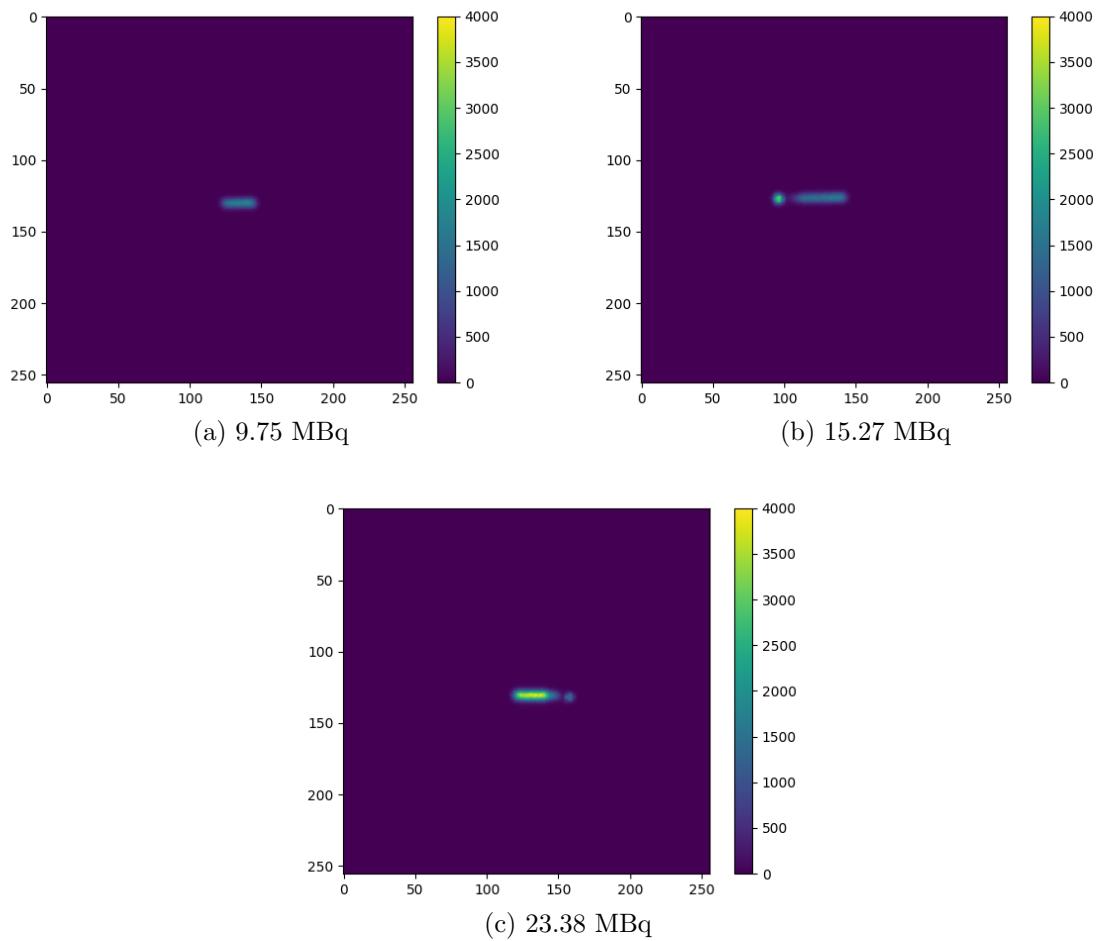
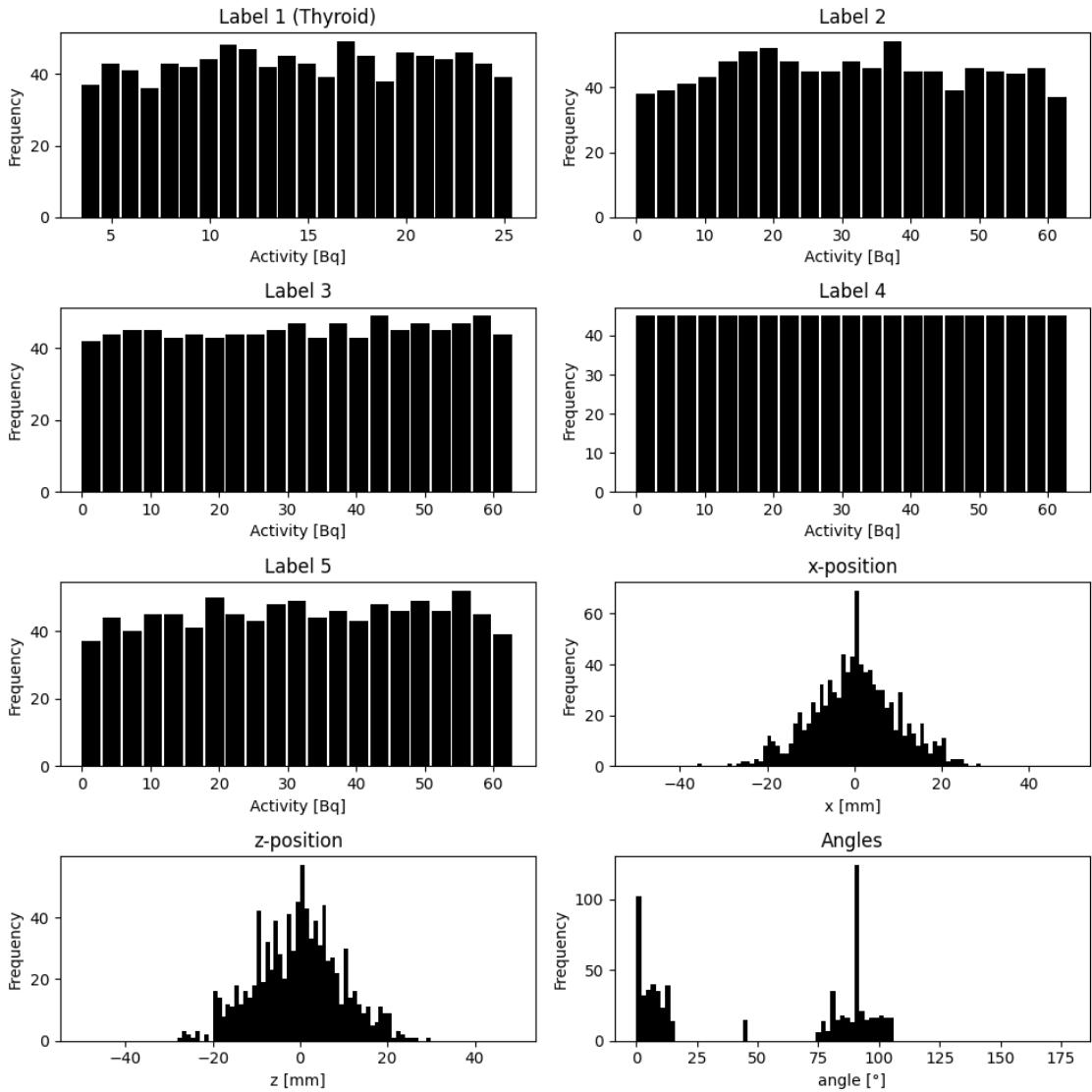


Figure 4.14: The three measurements taken with the gamma camera of the syringes.



*Figure 4.15: Histograms of the parameters in the dataset. The angles are all displayed in one histogram, even though they are from the different rotation axis.*

docker image named `korbiniant/gate_aws_ifl:1.5` was created and made publicly available to pull from the docker hub. The activity of the thyroid and nodules was uniformly sampled. For the positioning, roughly 230 samples were manually set, while the rest was sampled from a Gaussian distribution. Additionally, as in the positioning for the 230 samples, the angles were set manually and rotated around the y-axis by mostly 0 and 90 degrees. The distribution of the data is visualized in Fig. 4.15. However, as random data augmentation was used, the position and rotation of the dataset were different in the training. In theory, an infinite amount of samples can be generated. Only the parameters need to be specified in a CSV file. Afterward, the algorithm will use all available CPU cores and parallelizes the GATE-Rendering pair generation.



# Chapter 5

## Implementation

### 5.1 Ground Truth Generation with GATE

As explained in Section 3.3, for the simulation to run, the camera geometry, phantom geometry, digitizer (detector response), times parameters (start and stop times), data output format, physics processes, radioactive source(s) and verbosity have to be defined. Each of them will be explained separately in this section, except verbosity, as it does not change the final result but is for logging purposes only.

#### 5.1.1 Camera Geometry

In the simulations, the geometry was designed to model the *Mediso Nucline TH22* camera as accurately as possible. Two conflicting brochures of the camera with specifications were available, and the manufacturer was contacted to get as much information as possible. Unfortunately, the information from the manufacturer was very limited. Hence, the replication of the real-world measurements in GATE was performed via trial and error. The camera model geometry in this thesis is made out of a crystal, a collimator, and a 0.5 mm thin layer of aluminum in front of the crystal to model the cover of the camera. The PMTs and shielding were not included as they did not change the output image and increased simulation time significantly. The output format projection is used with  $256 \times 256$  pixels with  $0.9 \text{ mm} \times 0.9 \text{ mm}$  pixel size.

#### Crystal

For the scintillation crystal, information about the material and dimensions were available. It is made out of Thallium-doped Sodium Iodide ( $\text{NaI}(\text{Tl})$ ) and has dimensions  $230 \text{ mm} \times 6.5 \text{ mm} \times 210 \text{ mm}$ . In GATE, this was implemented via a box

volume with the NaI(Tl) material defined as a solid material with density  $d = 3.67 \text{ g/cm}^3$ , 15.2% sodium, 83.8 % iodine, and 1.0 % thallium.

## Collimator

There are several different collimators available for the camera. At Klinikum rechts der Isar, there were two collimators available: the LETH and HETH parallel hole collimators, which we assume to be hexagonal. For thyroid scintigraphies, the LETH (Low Energy THyroid) collimator is used. For the dimensions and material, there were conflicting declarations, but in the end, lead was used as the material, with a hole length of 35 mm, a hole size diameter  $D$  of 1.9 mm, and a septa thickness  $ST$  of 0.2 mm. The width and length of the whole collimator were chosen to be the same as the crystal. The collimator holes were implemented according to Section 3.3 with a box of lead with dimensions  $230 \text{ mm} \times 6.5 \text{ mm} \times 21 \text{ mm}$ , the hexagonal hole of air with radius  $R = 0.95 \text{ mm}$ , height  $H = 35 \text{ mm}$ , a repeating of 63 holes in the x-direction, and 99 holes in the z-direction.

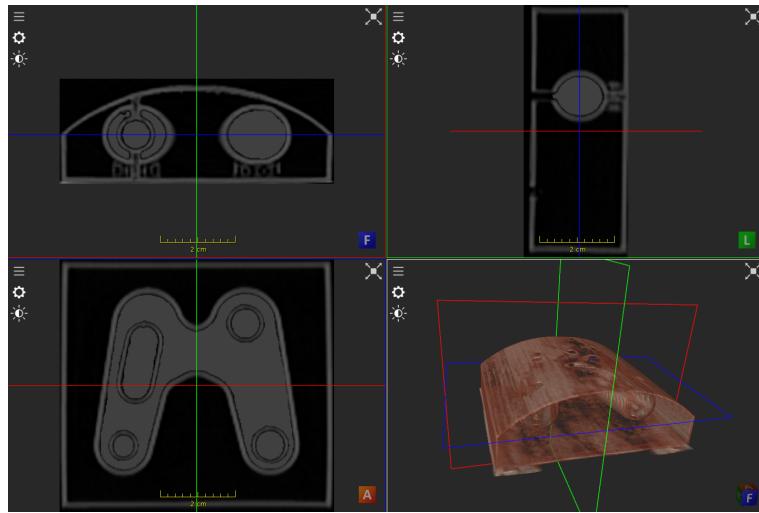
### 5.1.2 Phantom Geometry

The imported MHD file is the CT of the phantom acquired as explained in Section 2.1.4. All nodules inside the thyroid phantom were digitally filled with water, meaning  $HU = 0$ , because the  $^{99m}\text{Tc}$  is diluted in water. For this, labels made from this CT are used. The CT phantom and the phantom imported into GATE can be seen in Fig. 5.1.

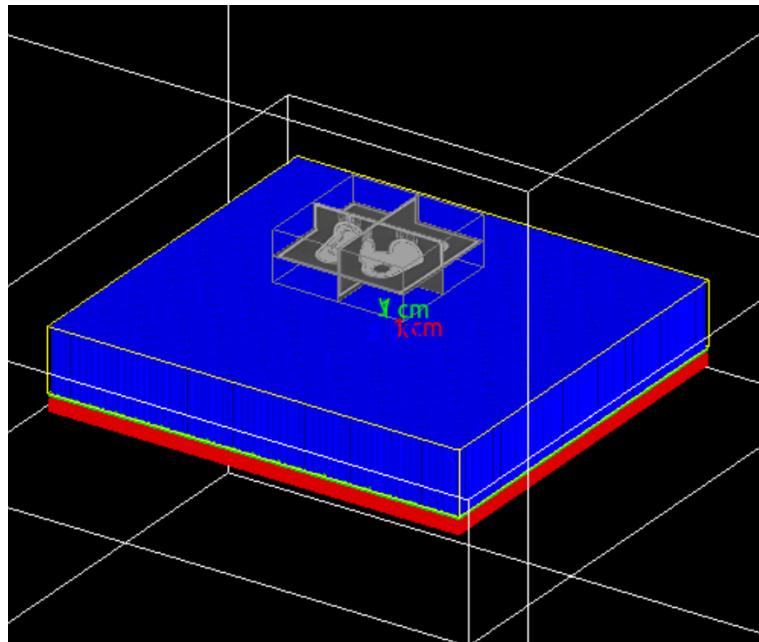
A custom materials conversion table was created as only three materials were used in the phantom: PETG, air, and water. The air material was assigned to all voxels<sup>1</sup> in the phantom from -2000 to -10, the material water from -10 to 1, and everything else was assigned PETG. The PETG solid material was custom created with a density of  $1.4 \text{ g/cm}^3$ , 14 parts carbon, 20 parts hydrogen, 5 parts oxygen, and one part sulfur. Finally, the position and rotation of the phantom are set accordingly. When compared to the real-world measurements, the optical tracking results from Section 4.1.2 are used. As the optical trackings coordinate system tracks the bottom middle of the phantoms plate with respect to the center of the collimator, they need to be transformed into the coordinate system in GATE. As GATE is using the center of the phantom as the origin, the tracked distance (z-axis) needs to be shifted by  $-13.89 \text{ mm}$  to be at the center of the phantom. Additionally, there is a difference of  $\Delta z = 31.25 \text{ mm}$  between the origin of GATE and the origin of the

---

<sup>1</sup>They have values in HUs.



(a) CT of phantom filled with water.



(b) The phantom imported into GATE together with the camera geometry. Only cross-sections of the phantom are visualized.

*Figure 5.1: Visualizations of the phantom geometry.*

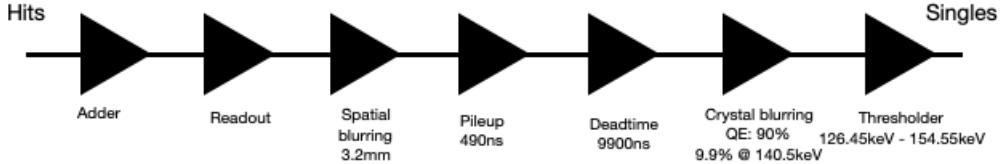


Figure 5.2: The whole digitizer chain visualized.

tracking coordinate system. This yields a total shift vector of

$$\vec{\text{shift}} = \begin{pmatrix} 0 \\ 0 \\ 17.36 \end{pmatrix} \quad (5.1)$$

On top of a shift the coordinate system of GATE is also extrinsically rotated by 90° around the y-axis and 90° rotated around the x-axis in this order. This transformation yields the rotation matrix  $\hat{\mathbf{R}}_{yx}$ :

$$\hat{\mathbf{R}}_{yx} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix} \quad (5.2)$$

The position  $\vec{\text{pos}}$  of the phantom in GATE is hence calculated with the optical tracking result  $\vec{\text{OT}}$  by:

$$\vec{\text{pos}} = \hat{\mathbf{R}}_{yx} (\vec{\text{OT}} + \vec{\text{shift}}) \quad (5.3)$$

The rotation of the phantom was read off manually from the optical tracking and put into GATE. In the future, the rotations can be easily included in the previous calculations using 4×4 matrices if it needs to be automated.

### 5.1.3 Digitizer

The *adder*, *readout*, *spatial blurring*, *pileup*, *deadtime*, *crystal blurring*, and *threshold* modules were implemented. According to the information available, the spatial resolution was set to 3.2 mm, and the energy blurring was set to 9.9% at 140.5 keV. The pileup and dead time were set to 490 ns and 9900 ns (with the paralysable mode), respectively. For the quantum efficiency in the crystal blurring module, 90% was chosen via trial and error. The threshold set an energy window of 126.45 keV - 154.55 keV. The whole digitizer chain is visualized in Fig. 5.2.

Energy [MeV]	Fraction of emissions
0.0869	0.000014
0.140511	0.885
0.142675	0.00023

Table 5.1: Gamma emission spectrum of  $^{99m}\text{Tc}$  [11].

### 5.1.4 Time Parameters

For the time parameters,  $t = 0$  s was used for the start time and simulated for ten seconds until  $t = 10$  s. In reality, thyroid scintigraphies are taken for 300 s. As this was not feasible to simulate in a decent time frame, it was decided to simulate only for 10 s as this was sufficient to get enough hits in the detector. Additionally, it was seen that the simulations crashed after exactly 2147483647 events. When simulating with high activities and 300 s this number of events is reached. This number is not random, as we figured out this is the highest representable number of integers stored in 32 bit<sup>2</sup>. Hence, the number of events is stored in a 32 bit integer format and causes the crash once this limit is reached.

### 5.1.5 Physics

In our case the `emstandard_opt4` physics list is used as this is the standard physics list for low-energy medical physics applications. Radioactive Decay was not used due to unexpected behavior (see Section 6.1.1).

### 5.1.6 Source Definition

The source is defined with the labels made from the CT. It is a 3D image saved in the MHD format. A linear translator was manually defined, setting the scale value to 0.885244 Bq. This is used in the voxelized source definition where the imported image values are multiplied by the scale value. This means, for example, if a voxel has the value 1, GATE assumes 0.885244 Bq of activity inside that voxel. The default position of the voxelized source is in the 1<sup>st</sup> quarter. So the voxelized source has to be shifted over half its dimensions in the negative direction on each axis. In our case, this shift vector is (-36.412 mm, -13.887 mm, -33.5 mm). As  $^{99m}\text{Tc}$  is used as a source, a *gamma* source was defined in GATE with a half-life of 21636s. The gamma emission spectrum of  $^{99m}\text{Tc}$  was provided via a table depicted in Table 5.1. The sum of this spectrum yields 0.885244 used for scaling in the translator. Lastly, the emissions angles were not limited and set to *iso*. For parts of the parameter

---

<sup>2</sup>32bit = 1 bit sign+31 bits values.  $2^{31} - 1$ (because of 0) = 2147483647

tuning, a syringe filled with  $^{99m}\text{Tc}$ -pertechnetate was used and simulated. For that, the source was modeled as a cylinder with the syringes dimension and with the activity defined according to the measurement.

## 5.2 Rendering

To implement the raycasting algorithm, Python was used. As in the GATE simulations, the labels made from the phantoms CT are defined as the source and imported. Additionally, the CT of the phantom is imported in the same way. As a CT is given in HU, it is converted into attenuation values for the  $^{99m}\text{Tc}$  emission energy of 140keV via:

$$\mu(140\text{keV}) = \frac{(0.000155 \cdot \text{HU} + 0.155)}{10} \left[ \frac{1}{\text{mm}} \right] \quad (5.4)$$

These values can be derived using the x-ray attenuation coefficients of water and air at 140 keV.

As an orthographic projection is needed, the focal length is set to 29970 mm and the origin of the camera is at a distance of 30000 mm. This way, the rays are almost parallel when entering the volume. The image dimensions and image pixel size are set to 256 mm  $\times$  256 mm and 0.9 mm  $\times$  0.9 mm respectively, according to the GATE simulations and real-world camera specifications. The extrinsic matrix is composed by multiplying the rotations around the x-axis, y-axis, and z-axis in that order. To compensate for the difference between the GATE and renderings coordinate systems, an initial rotation of 270° around the x-axis and -90° around the y-axis is introduced. Afterward, the translation vector is multiplied by the final rotation matrix and used in the extrinsic matrix. This way, the distance can always be set with one dimension, while the x and y translation can be changed with the x and y translation parameters. To calculate the entry points and exit points for each ray/pixel, the vector  $\vec{r}$  is multiplied by the camera's rotation matrix, where  $r$  and  $c$  are the rows and columns indices of the pixels in the image plane,  $f$  the focal length,  $[w_x, w_y]$  the pixel dimensions, and  $[x_0, y_0]$  the center of the image.

$$\vec{r} = \begin{pmatrix} \frac{w_x(r-x_0)}{f_x} \\ \frac{w_y(c-y_0)}{f_y} \\ 1 \end{pmatrix} \quad (5.5)$$

An efficient bounding-box vs. ray collision algorithm [75] is used to calculate the

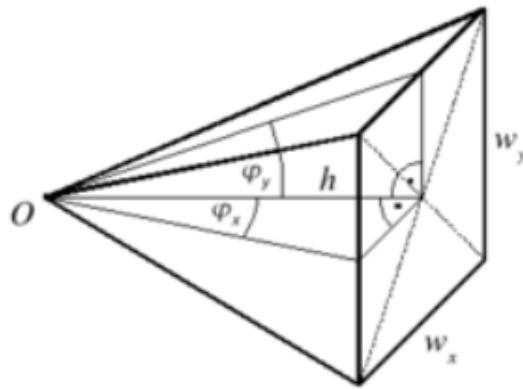


Figure 5.3: Solid angle of a pyramid [73].

intersection of these rays with the volume. This algorithm yields the entry points and exit points of each ray/pixel of the image with the volume. The custom shader calculates the final pixel values the same way for every pixel/ray using its entry points and exit points. It works as the following: First, it samples along the ray on the label volume with nearest-neighbor interpolation until a non-zero value is sampled. This means the ray has hit a voxel in the source label map. If this happens, the sampling steps are reversed and on the way back, the values of the transformed CT are sampled and the original value of the label is attenuated via the Lambert-Beer-law (see Eq. 2.5) in the predefined step size. This is repeated until the entry point of the volume is reached. Additionally, a solid-angle ratio is calculated based on the distance traveled and the distance from the entry point to the image plane. This percentage is multiplied by the attenuated value. That step is crucial because the activity in the voxel is defined isoangular, meaning if one decay happens inside this voxel, the gamma-ray is emitted equally likely in all directions. If this step would be left out, one would assume every gamma ray would fly toward the camera, resulting in overestimating the activity.

The percentage of rays hitting the camera depends on the distance of the source to the camera, and the area one wants to calculate the solid-angle for. In this project, a solid angle is calculated using the distance to the imaging plane and the pixel size as the area. For the percentage, the solid angle is calculated by:

$$\Omega = 4 \cdot \arctan \left( \frac{w_x \cdot w_y}{2 \cdot h \cdot \sqrt{4 \cdot h^2 + w_x^2 + w_y^2}} \right) \quad (5.6)$$

and then divided by  $4\pi$ , where  $h$  is the distance, and  $w_x$  and  $w_y$  are the pixel sizes (see Fig. 5.3). This value is saved, and the sampling on the source label volume continues until it hits another non-zero voxel. Then the same reverse sampling is performed and the yielded value is summed with the previously saved value. This

is repeated until the end of the volume is reached and all values are summed up. Additionally, on every sampling step for the labels, it is checked if this voxel has already been sampled and used for the reverse sampling. This is important because depending on the sample step size, the angle of the ray with respect to the voxel, and the pixel size (equivalent to the distance between rays), it can happen that several rays sample the same voxel activity. Once an activity is sampled and used for the calculation, this gamma ray is already used, and can not be used for other rays anymore.

To understand it more easily, imagine a voxel with dimensions  $1 \text{ mm} \times 1 \text{ mm} \times 1 \text{ mm}$  and activity  $1 \text{ Bq}$ . When one perpendicular ray comes from a camera with the same pixel size as the voxel size of the volume and samples that voxel, it registers one decay<sup>3</sup>. Now when the pixel size decreases by half, four rays sample that voxel and every ray registers one decay. This means in total there would be 4 decays happening when in reality there is only one decay. The same applies if the step size of the sampling is reduced to a lower length than the voxel dimensions. This is a common problem when working with nuclear medicine data, especially with activity volumes, where changing the voxel dimensions will also change the total activity of the volume.

As GATE simulates for ten seconds, the rendering has to be adapted by multiplying the result by 10. This is because the activity is defined in  $\text{Bq}$  or  $\frac{1}{\text{s}}$ , meaning the initial rendering result would only be valid for  $1 \text{ s}$  simulations.

Finally, if a pixel has a value of one, it means that on average, one ray will hit the detector in ten seconds. If the value is 0.5, one hit will only be expected if 20 s are simulated. Of course, hits can only be integer values, but as decays are stochastic, one can continue to work with these floating-point numbers.

### 5.3 Deep Learning

**Normalization** Normalizing the whole dataset between 0 and 1 made training significantly more stable. To keep the relative differences between the images, a min-max normalization was implemented. A scaling factor  $s$  was calculated by:

$$s = \frac{1}{max - min} \tag{5.7}$$

where  $max$  and  $min$  are the maximum and minimum pixel values of the whole dataset. During data-loading the  $min$  value is subtracted from the input and target

---

<sup>3</sup>Assuming no solid angle for now.

images<sup>4</sup> followed by multiplying both with  $s$ .

**Data Augmentation** Random rotations and translations were implemented with random affine transformations. To make sure that the translations are not out of the image dimensions, the maximum and minimum coordinates of non-zero pixels in both x and y directions were calculated on all input images. Additionally, a safety margin of 20 pixels was applied in both dimensions. The rotations were randomly sampled between  $-30^\circ$  to  $30^\circ$  around the y-axis. In every training step, a new random transformation is applied, so the model sees new images every step.

**Metrics** To evaluate the models' performance several metrics were introduced. The Peak Signal-To-Noise Ratio (PSNR), the Structural Similarity Index Measure (SSIM), and a custom metric called Relative Pixel Sum Loss (RPSL). They are defined as:

$$PSNR(I, J) = 10 \cdot \log_{10} \left( \frac{\max(I)^2}{MSE(I, J)} \right) \quad (5.8)$$

$$SSIM(I, J) = \frac{(2\mu_I\mu_J + c_1)(2\sigma_{IJ} + c_2)}{(\mu_I^2 + \mu_J^2 + c_1)(\sigma_I^2 + \sigma_J^2 + c_2)} \quad (5.9)$$

$$RPSL(I, J) = \frac{|\sum I - \sum J|}{\sum I} \quad (5.10)$$

where  $I$  and  $J$  are the two images to compare,  $\mu$  the mean of the image,  $\sigma$  the standard deviation of the image,  $\sigma_{IJ}$  the covariance between two images,  $c1$  and  $c2$  constants used to numerically stabilize the division, and  $\sum I$  the sum of all pixel values in the image.

**Training & Hyperparameter Tuning** To train the network, several hyperparameters were investigated. As activation functions, the tanh and PReLU functions are used in the sweep:

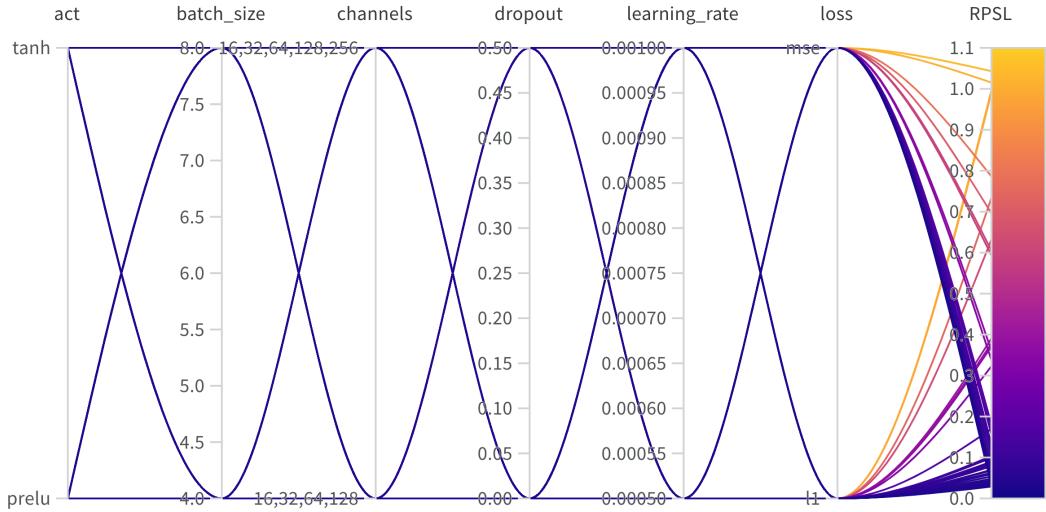
$$f(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (5.11)$$

$$f(x) = PReLU(x) = \begin{cases} x & \text{if } x > 0 \\ a \cdot x & \text{otherwise} \end{cases} \quad (5.12)$$

where  $a$  is a learnable parameter that allows the activation function to have a slope for negative input values. Two different batch sizes of 4 and 8 samples per batch were investigated. The depth of the U-net was changed in two versions as well: (16,

---

<sup>4</sup>In our dataset the min value was 0, so it did not make a difference.



*Figure 5.4:* Hyperparameter tuning of the model using sweeps. Hyperparameters are the activation function *act*, the batch size *batch\_size*, depth of the U-net *channels*, *dropout*, the learning rate *learning\_rate*, and the loss function. The goal was to minimize the Relative Pixel Sum Loss (RPSL).

32, 64, 128) and (16, 32, 64, 128, 256) where each number represents the number of channels in the current layer. Dropout values of 0.5 and no dropout, learning rates of 0.001 and 0.0005, and two different loss functions were tested. The two losses Mean-Squared-Errors (MSE) and L1 are defined as:

$$MSE(y, \hat{y}) = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (5.13)$$

$$L1(y, \hat{y}) = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (5.14)$$

where  $y$  is the ground truth,  $\hat{y}$  is the predicted image, and  $n$  is the number of samples. All 64 combinations of these hyperparameters were trained for a maximum of 1000 epochs with an early-stopping callback on the validation loss with a patience of 100 epochs. The goal was to minimize the RPSL metric (see Fig. 5.4). Afterward, the best five models were trained for a maximum of 2000 epochs and evaluated on the test dataset.

**Final Model** The best model with respect to RPSL of the five best runs of the sweep has the parameters depicted in Table 5.2. It was trained for 540 epochs for 25 minutes before the early stopping callback stopped the training. The model has 659K trainable parameters.

Activation function	PReLU
Batch size	4
Depth	(16,32,64,128,256)
Dropout	0.0
Learning rate	0.0005
Loss function	MSE

Table 5.2: Hyperparameters used in the final model.

## 5.4 Iteration Loop

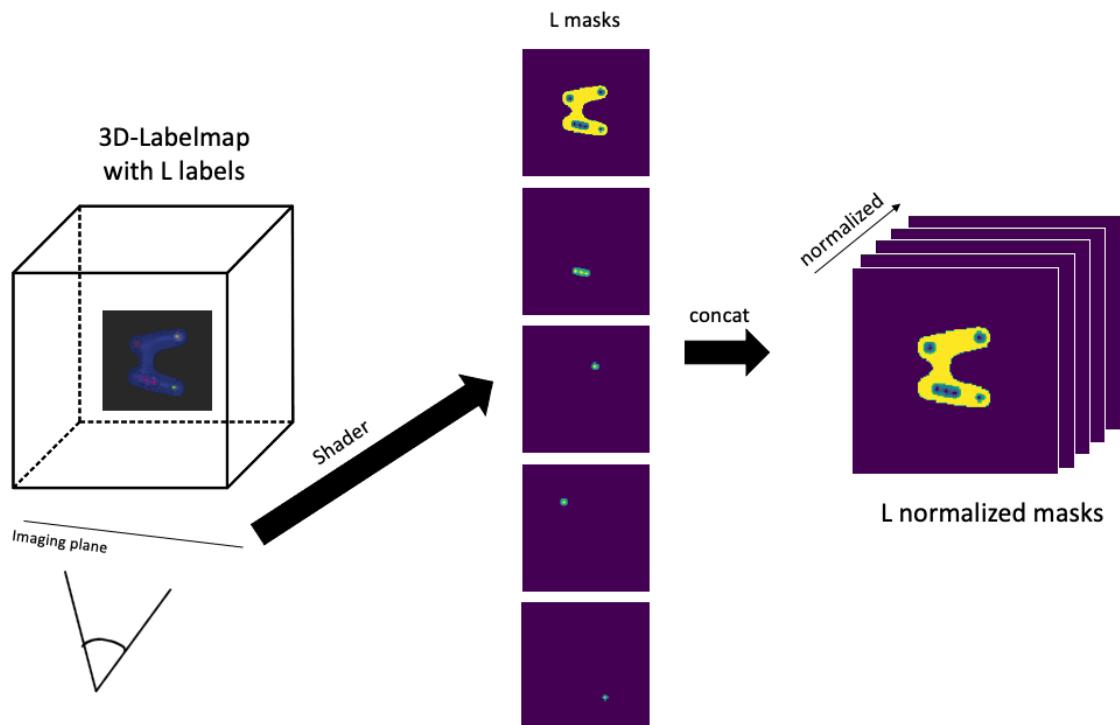
To update the change per voxel in each label, we need the 3D label map, the complete set of parameters to simulate the rendering, the current image, and a reference image<sup>5</sup>.

First, the raycasting algorithm is used to project the 3D label map on a 2D image using the same camera model and pose used in the rendering. However, a different shader is implemented to convert the sampled values along the ray into pixel values. In this case, the number of sampled values per label is counted, generating one count number for each label per pixel. The 2D projection of the label map is expanded into a 3D array with dimensions  $256 \times 256 \times L$ , where  $L$  represents the number of unique labels. Each channel in the 3D array represents the count number of one specific label. This represents a 2D mask for every label, meaning only non-zero pixels are affected by the voxels along the ray. The array is then normalized along the channel axis to obtain relative values for each label, representing the percentage of labels sampled on the ray for each pixel. These  $L$  masks must be calculated once at the beginning of the iteration loop (see Fig. 5.5).

Finally, these masks for each label are multiplied with the current image and the reference image separately, yielding  $L$  images for the current and the reference image. All pixel values of each image are summed up independently, yielding one value for each label for the current and one value for each label for the reference image. These  $L$  values are saved in a separate list for the current and the reference image. Then, the difference between these two lists of length  $L$  is calculated and divided by the total amount of voxels for each label in the 3D-label map. This yields one scalar for each label, which represents the difference in the total amount of hits between the target and current image each label voxel contributes. However, to update the activity a proportionality factor  $f$  is needed, that connects the number of hits in the target image each label voxel contributes, to the inverse, meaning the activity that each voxel has to change to get matching images.

---

<sup>5</sup>in our case the real scintigraphy or GATE



*Figure 5.5: Algorithm to calculate  $L$  masks for the iteration loop. Step 1: Project  $L$  labels on the imaging plane. Step 2: Expand projection into  $L$  channels where each channel represents the count number of this label on the ray. Step 3: Normalize along the channels to get relative percentages of the labels. These masks have to be calculated once at the beginning of the iteration loop. Afterward, they can be reused in each iteration step.*

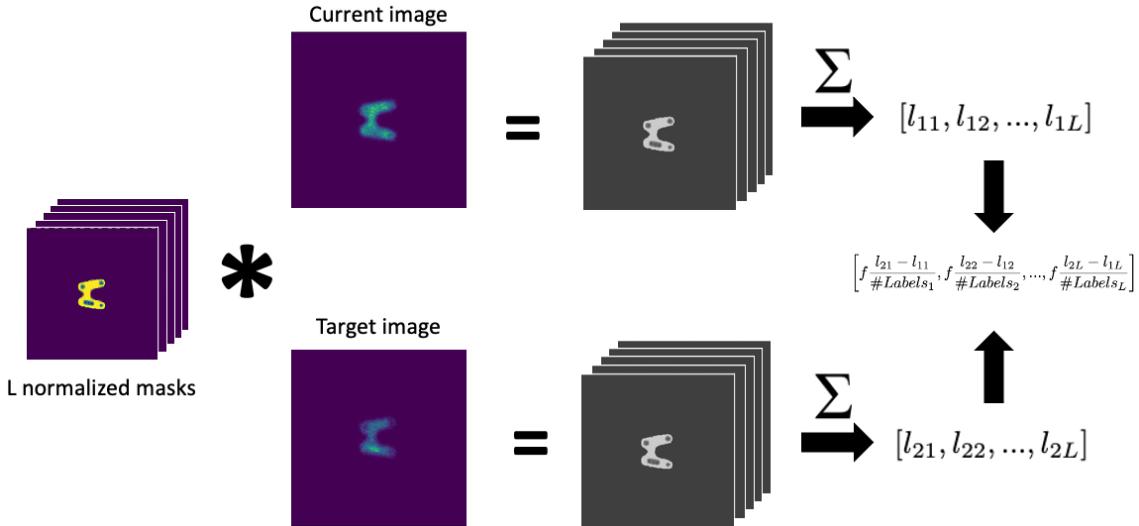


Figure 5.6: Algorithm to calculate the change per voxel per label, given the masks. Step 1: The mask for each label is multiplied with the reference and current image, yielding  $L$  new images for the current and the reference image. Step 2: All pixel values in each yielded image are summed up and put in a list, yielding a list with  $L$  elements for the reference and the current image. Step 3: The difference between these two lists is calculated element-wise and divided by the amount of voxels the initial 3D-Labelmap has for this label. Additionally, all elements are multiplied by a proportionality factor  $f$ .

This yields one final value per label, which is the change of activity per voxel in that label, for the update of the 3D-activity map for the next iteration step (see Fig. 5.6). Additionally, the choice of the proportionality factor  $f$  and a learning rate decay was investigated. Finally, the factor  $f$  was set to 1500 and no learning rate decay was introduced.



# Chapter 6

## Results

### 6.1 GATE

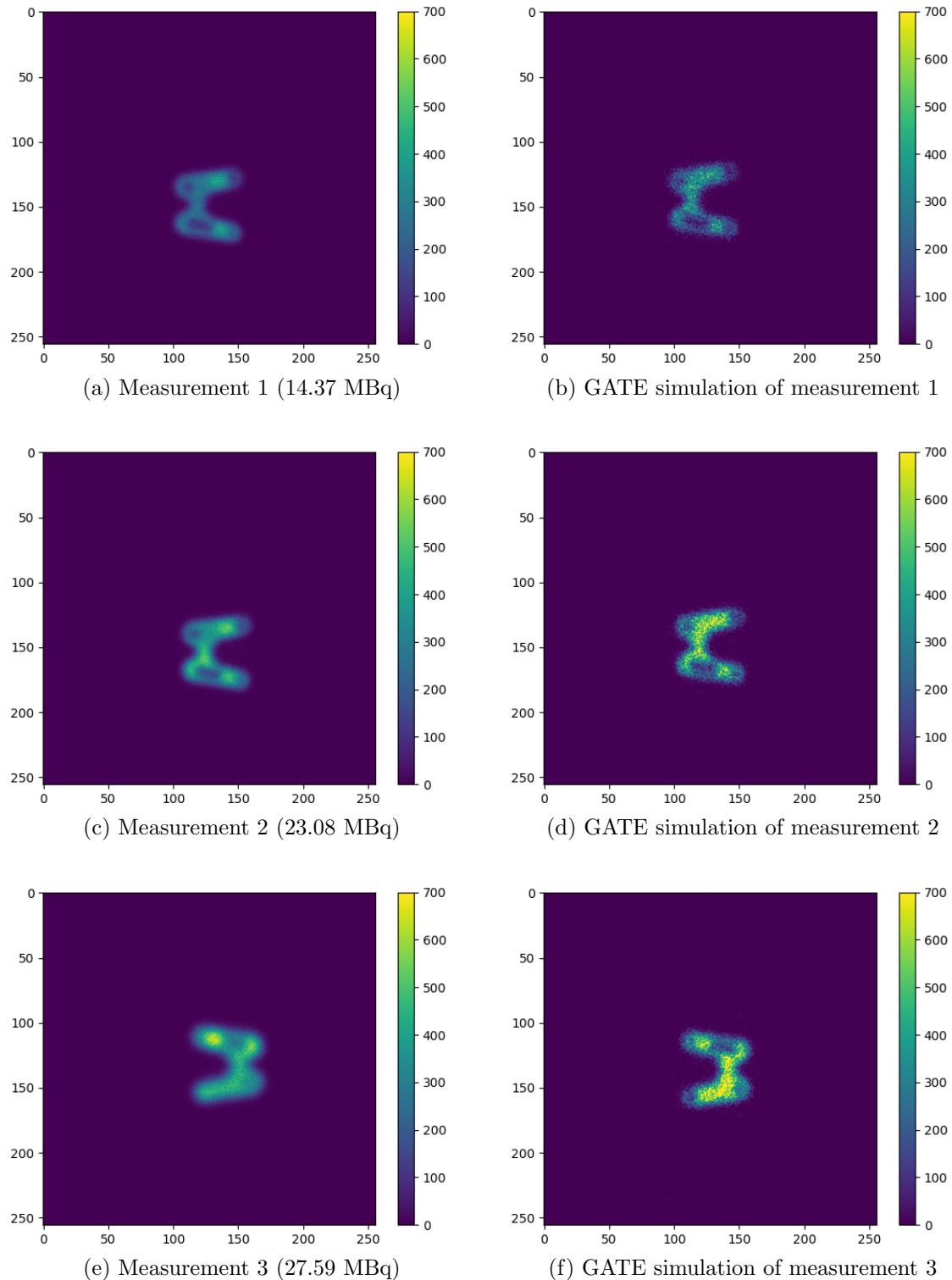
The GATE simulations were acquired for 10 s and the parameters described in Section 5.1 are used. The three measurements of Table 4.1 were reproduced in GATE. As only 10 s were simulated but the measurements were acquired for 300s, the result was multiplied by 30. The results can be seen in Fig. 6.1. Additionally, the total amount of hits in the image was investigated and compared. A summary of the hits in the three experiments can be seen in Table 6.1.

#### 6.1.1 Discussion

It can be seen in the results, that the GATE simulations do not perfectly match reality with respect to count rate. In fact, the count rate of the images was investigated the most in this thesis. There are many possible reasons for this difference. Primarily, this was no optical simulation, meaning not every photon produced by the scintillation crystal is simulated. Thus the GATE simulation is already a simplification. The simplification was done because simulating optical photons increases the simulation time by a factor of 1000, which is not feasible.

Experiment	Experiment 1	Experiment 2	Experiment 3
Hits in Measurement	443305	627488	844040
Hits in GATE	424440	664560	790230
Rel. difference	-4.3%	+5.9%	-6.4%

Table 6.1: Summary of the total number of hits in the image. The relative difference is calculated via  $\frac{GATE - Measurement}{Measurement} \cdot 100$ .



*Figure 6.1: The three measurements of the phantom together with its reproduced simulation in GATE. In the left column are the measurements of the three phantoms. The right column entails its corresponding simulation in GATE*

Source type	Mono energetic	Unscaled spectrum	Scaled spectrum without alu	Scaled spectrum with alu	Ion source
hits	4453	4418	3948	3906	3919

Table 6.2: Summary of hit rates for different source types.

**Camera materials** Moreover, as not every part of the camera is known, it is possible that some internal parts of the camera are not included which influences the total hit number. To compensate for this the best, the 0.5mm aluminum cover in front of the crystal was included. Many different materials were considered, such as a tungsten aluminum alloy as suggested by [43]. However, after contacting different vendors of crystals, none of them were aware of such a tungsten alloy. Hence, this tungsten alloy was not implemented. There would also be no legitimate reason in our knowledge to use such a highly attenuating material as a cover because many rays would be absorbed.

For the collimator material, the most used collimator materials are tungsten and lead. The manufacturer gave conflicting answers on which is used in the camera. Hence, simulations were conducted with both materials to investigate the difference. The lead collimator yielded 4.0% more hits than the tungsten collimator. However, according to the manufacturer, the collimator is made with *microlinenar technology*, which is mainly used with lead [22]. In addition, information from the manufacturer that lead is the material of collimators is more recent than the information that they use tungsten. Hence, lead is used as the final collimator material. As it is not certain if this was really used as the collimator material in the camera, that could explain parts of the differences.

**Source types** Various source types were investigated, including monoenergetic gamma (140.5keV), unscaled spectrum, scaled spectrum with 0.885 (with and without aluminum cover), and ion source with radioactive decay (see Table 5.1). The ion source, which simulates a radionuclide based on its atomic number, atomic weight, ionic charge, and half-life, is the most realistic but slowest. Identical simulations were conducted with a low activity and 10 s simulation time to speed up the process. Results are shown in Table 6.2, and it can be observed that the scaled spectrum with aluminum cover closely resembles the realistic ion source with minor differences and is five times faster<sup>1</sup>. For these reasons, the scaled spectrum with the aluminum cover was used.

---

<sup>1</sup>1h 12m compared to 6h 42m

**Digitizer** The digitizer and its components were carefully investigated. The *adder* module can not be modified and hence no parameters were investigated. The *readout* module describes the readout policy between multiple crystals. As the used camera only has one crystal, this module was also not further departed from the default values. The single crystal was also the reason why several other digitizer modules, such as the *local efficiency* module were not included. The spatial blurring parameters were kept constant at their defaults, as this was specified by the manufacturer and implemented accordingly. The *pile-up* and *dead time* modules were included as pileup can affect hit rates. It can be described by the detector's response time it takes to form the signal after a ray arrives. During this time, if a second event is detected it will pile up on the first. The resulting pulse is then a combination of the two rays. Deadtime is due to the shaping time of signals or any other reason. During that time, subsequent hits detected on the same electronics can hide. In total, nine different combinations of pile-up and dead time were simulated and evaluated. The results can be seen in Table 6.3. The results show

		PU		
		not included	490ns	4900ns
DT	not included	98005	98150	97686
	990ns	97948	98119	97686
	9900ns	97476	97586 (used)	97405

Table 6.3: Summary of hit rates for different pile-up and dead time combinations.

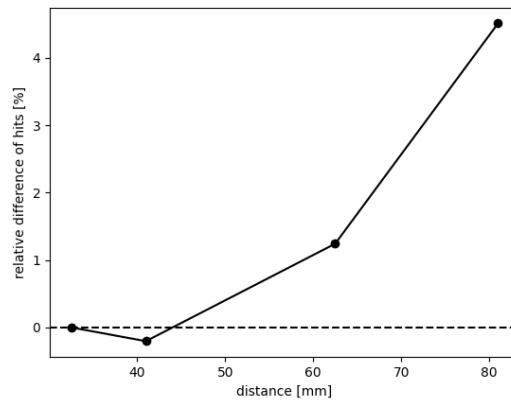
no significant change in hit rates by the two modules. This is because they play a role in higher activities, as more hits are registered in a certain time frame. Finally, a dead time of 9900 ns and pile-up of 490 ns is used, as these values are used for other SPECT devices in [40] and [65] respectively. The *thresholder* module to set the energy window was also not changed, as the energy window was specified in the measurements. The most significant change in hit counts has been achieved with the *crystalblurring* module because the quantum efficiency of the crystal can be specified. The default value for the quantum efficiency is 100%, meaning every event is detected by the photo-detector. Therefore, this parameter represents the effect of the transfer efficiency of the crystal and of the quantum efficiency of the photo-detector. Hence, this quantum efficiency value correlates with the hit counts. If the quantum efficiency would be 50%, half of the hits would not be counted. To match the hit counts in the final image, this parameter can be adjusted. To get the best result for the three measurements, the value was set to 90%. Additionally, this module implements energy blurring. A range of percentages and a reference energy can be set. The module randomly sets the energy blurring in that range. As in

the camera used, only one crystal is used and only one energy blurring resolution is specified, this range of percentages was set to one value of 9.9% and a reference energy of 140.5 keV.

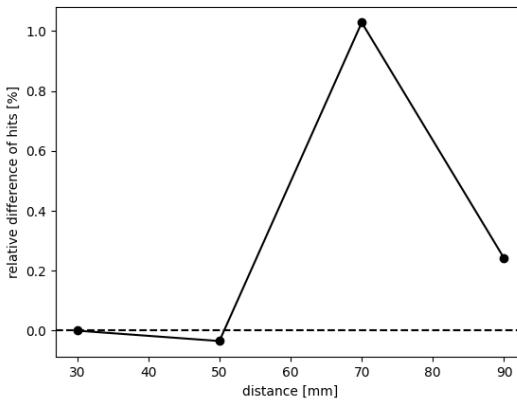
**Physics parameters** For the physics parameters, adding processes manually and comparing them to physics lists, which are a collection of processes, was briefly investigated but only minor or no differences were detected. Finally, the `emstandard_opt4` is used as it uses the most accurate models for low-energy particles [23]. The *RadioactiveDecay* module was not included in the digitizer as it showed unexpected results. If the measurements, were taken a significant time after measuring the activity, a lower activity is expected as some parts of the radionuclide already decayed. Simulating for one second with starting time  $t = 0\text{s}$  yielded reduced count rates when the radioactive decay module is included, even though one would expect no significant change in activity in the first second. We do not know the reason for this behavior, resulting in excluding this module. If the ion source is used, the *RadioactiveDecay* module has to be activated.

**Errors in measurements** It should be noted that there can also be intrinsic errors in the measurements. First of all, the measurement of the injected activity with the *ISOMED 2010* activimeter has a basic error of  $\pm 5\%$ . Additionally, the filling could have had some inaccuracies with minor leftovers in the needle of the syringe. The leftovers of  $^{99m}\text{Tc}$  in the syringe were measured again in the activimeter after filling the phantom, but the needle was not included, as it does not fit into the device. Additionally, as only one 3D-printed phantom was checked in a CT for leakage, it can not be certainly said that all phantoms were 3D-printed with sufficient quality to be leak-proof inside. This can influence the distribution of  $^{99m}\text{Tc}$  inside the phantom, and finally the resulting image. The last possible error we are aware of is some potential post-processing of the camera software which takes place after acquiring a measurement. We are not aware of some post-processing steps included in the measurements, but as only the final image can be retrieved, it is not certain.

**Distance of phantom** Additionally, the influence of the distance of the source to the collimator with respect to hit counts was investigated. We expected a decline in hit rates according to the solid angle, given in Eq. 5.6. However, different results were obtained in GATE simulations. A spherical source and the phantom were



(a) Phantom source



(b) Spherical source

*Figure 6.2: Hit counts of GATE simulations with different distances and sources.*

used as a source for simulations with different distances to the collimator. The hit counts can be seen in Fig. 6.2. Surprisingly, the hit counts slightly increased with the distance for the phantom source. For the spherical source, the hit counts varied only slightly but no trend of decline is visible. This finding will be discussed for the rendering results. In reality, there is also scattering, which influences the number of rays hitting the crystal. Due to time constraints, the amount of scattering in this experimental setup and in GATE was not further investigated.

To conclude, the found parameters for the GATE simulation match with the real-world measurements within a maximum difference of 6.4% of hit counts. Given the assumptions and limitations, we consider this result sufficient for further use in this thesis.

Time start	0s
Time stop	10s
Thyroid activity	6.0Bq
Nodule 2 activity	40.0Bq
Nodule 3 activity	0.0Bq
Nodule 4 activity	40.0Bq
Nodule 5 activity	20.0Bq
X-Translation Phantom	-5.0mm
Y-Translation Phantom	43.99mm
Z-Translation Phantom	-5.0mm
Rotation vector	0 1 0
Rotation angle	90

Table 6.4: Parameters used in the GATE simulation and the input for the rendering script. Activities are given per voxel.

## 6.2 Rendering

The renderings are calculated with the same parameters as the GATE simulations. Hence, they should produce similar results. The result of GATE and the rendering in this section are produced with the parameters seen in Table 6.4.

**Position and size** To compare the position of the GATE simulation and the rendered projection, binary masks for each image are created by assigning a value of one to non-zero pixels, and a value of zero otherwise. Finally, the two masks are subtracted from each other to get an image where the pixels are visible that show a difference between the GATE simulation and the rendering with respect to position and size (see Fig. 6.3).

One can clearly see that position and size of the rendering match accurately with the GATE simulation. Scattering and non-parallel rays are not implemented into the shader, resulting in an edge around the phantom. The position and size have to perfectly match with GATE because the rendering is also used in the iteration loop later, where an offset in position or size influences the performance significantly.

**Hit counts** In the example of Fig. 6.3, GATE counts 10601 hits, when the rendering only counts 6571 hits. The L1-loss of this sample is 5063.227, which is the sum of Fig. 6.3 (d). In the rendering script, the hit counts are mostly influenced by the distance of the volume to the camera as the solid-angle (see Eq. 5.6) is implemented. However, in Fig. 6.2, the dependence of the distance appears to be minor in the GATE simulations. As the solid angle had to be implemented to not overestimate the rays hitting the detector, the distance in the rendering and at the GATE

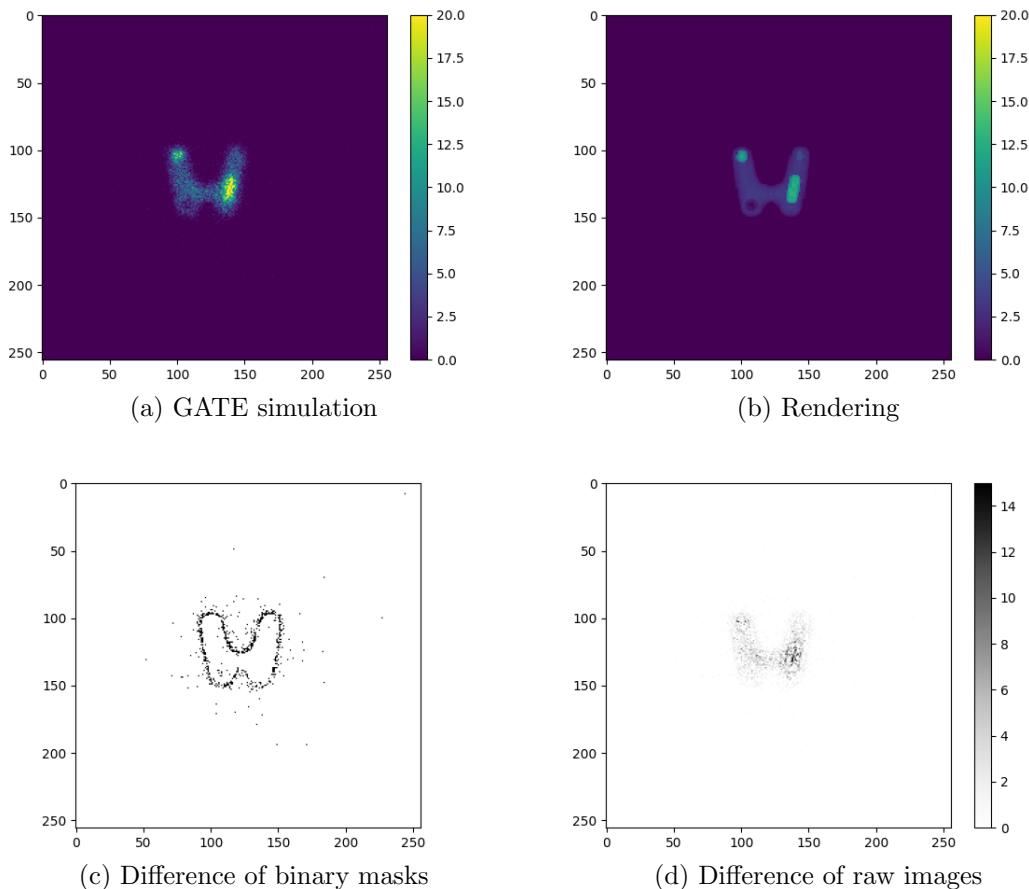
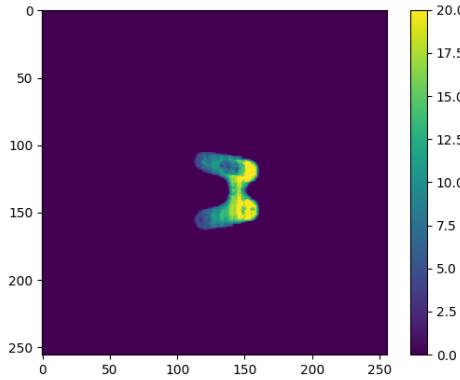


Figure 6.3: The rendering (b) compared to the GATE simulations (a) depicted for one sample. The difference of the binary masks of the two images is visualized in (c) and the difference between the raw images is visualized in (d).



*Figure 6.4:* Example of a rendering where the phantom was rotated so that one side is closer to the detector. The parts closer to the detector have a higher count rate as the solid angle is bigger.

simulations were kept constant for all simulations and renderings, that is 43.99 mm in GATE and 30 mm in the rendering. This way, the neural network should learn the difference and adjust it accordingly. Nevertheless, in the rendering, the 30 mm distance is defined at the middle of the phantom. If the phantom is rotated, parts of the phantom might be closer to the camera, and hence the hit counts will increase on these projected parts. An example of this is depicted in Fig. 6.4, where the phantom is rotated by 30 degrees, so that one side is closer to the detector, resulting in a bigger solid angle.

### 6.3 Deep Learning

Our final model (see Table 5.2) was evaluated on three metrics: PSNR, SSIM, and RPSL. These metrics were calculated on a test dataset which represents 10% of the whole dataset (see Table 6.5). The same sample used in Section 6.2 is taken for

$\uparrow$ PSNR	38.5289
$\uparrow^1$ SSIM	0.9788
$\downarrow$ RPSL	0.0347

*Table 6.5:* Evaluation metrics results on the test set of the final model. The arrows depict if the metric should be small or high.  $\uparrow^1$  means that the metric aims to be 1.

a comparison between the neural network and the rendering. Fig. 6.5 shows the GATE simulation, the corresponding rendering, and the corresponding output of the neural network.

The GATE simulation counts 10601 hits, the rendering 6571 hits, and the neural network output 10983 hits. RPSL between the rendering and GATE and the neural

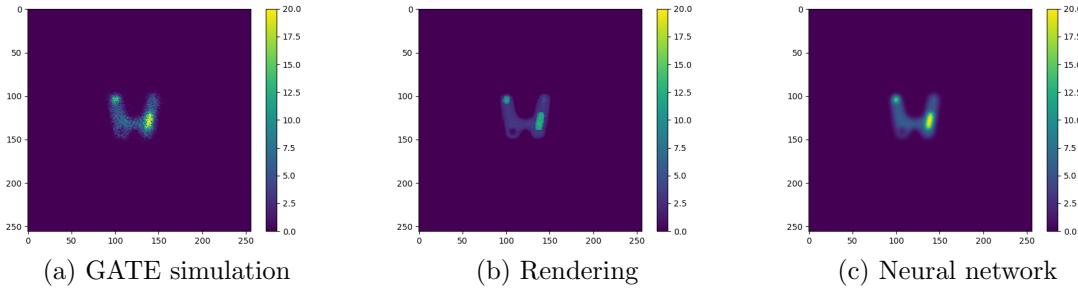


Figure 6.5: GATE simulation (a), rendering (b), and the output of the neural network (c) made with the same parameters.

	GATE	Rendering	Neural Network
Hits	10601	6571	10983
RPSL	0.00	0.380	0.036

Table 6.6: Comparison of GATE, rendering, and the neural network with respect to RPSL. An improvement in the RPSL of the neural network is clearly visible. As the GATE image is the reference, it is matching perfectly by definition.

network output and GATE are 0.380 and 0.036. This clearly shows the improvement of the neural network compared to the rendering (see Table 6.6).

## 6.4 Iteration Loop

### 6.4.1 GATE

The iteration loop was evaluated by simulating a GATE scintigraphy with a specified 3D activity map to generate the reference image. Another GATE scintigraphy is generated with an initially specified 3D activity map, which serves as our initial image. The iteration steps as described in Section 5.4 are performed. The calculated change in activity per voxel per label is added to the initial 3D activity map and the resulting GATE simulation is used as the current image for the next iteration step. As we specified the reference image, we know the ground truths of label activities and can compare them at each step for evaluation purposes. This information is not available to the algorithm.

**Proportionality factor  $f$  and learning rate** To set the factor  $f$ , several experiments were conducted. Initially, the solid angle calculation was used again to determine  $f$ . This solid angle yields the ratio of hits on the detector to the decay/activity. Hence, by using the inverse of the solid angle, this factor converts the hits-activity ratio into the activity-hits ratio. To do this, the solid angle ratio

for every voxel in the volume is calculated. The factor  $f$  is then calculated separately for each label by the mean of the inverse of the solid angle of all voxels that have that label. However, after implementation, this approach was discarded as it strongly depends on the solid angle again which is not reflecting reality correctly, resulting in values of  $f$  several orders of magnitude off from the desired value. This is fatal, as too big update steps result in extreme activities which take too long to simulate or all activities updated to negative activities which are clipped to 0.0 Bq. For this reason, we also experimented with introducing a learning rate decay, which dampens the update steps in case they oscillate around the ground truth.

Additionally, the factor  $f$  was set manually by experimenting with different values. However, the learning rate decay was finally not used, as no oscillation around the ground truth was observed. To set  $f$ , only one label was observed for the update step with different starting values and factors  $f$  (see Fig. 6.6).

It can be seen in all plots that the further away the starting value from the ground truth (dashed line), the bigger the update step in the correct direction. This validates the correctly working algorithm and that the learning rate decay is not needed. The figures also show that  $f = 2000$  looks like a good candidate because after one update step, all starting values converge to the ground truth. However, we prefer the algorithm to undershoot than to overshoot, which can be realized by choosing a smaller factor. Indeed,  $f = 1000$  shows this behavior but would take too long to converge. This trade-off between overshooting and time was solved by taking the factor  $f = 1500$ , which is in between.

**Different starting values and multiple labels** To evaluate the performance of the iteration loop using only GATE simulation, several experiments were conducted. The performance metric was manually defined by taking the sum of the residuals (SOR) between the ground truth and the current iteration activity values for each label. In the first experiment only two labels (the thyroid/Label 1 and one additional label), were given an activity, and all the other labels were set to 0 Bq (see Fig. 6.7). The same activity of 25 Bq was set for all five labels. It is clearly visible that the iteration loop is working as desired, as values converge to the ground truth. Additionally, the SOR decreases significantly. Excluding the calculation of the ground truth, this convergence took 33.41 hours. The duration of the simulation depends mostly on the simulated activity. Additionally, the simulation duration varies depending on the performance of the CPU.

In the second experiment, all labels were set to active, but overall low activity was chosen. This is to assess the iteration loops' performance for low activities where

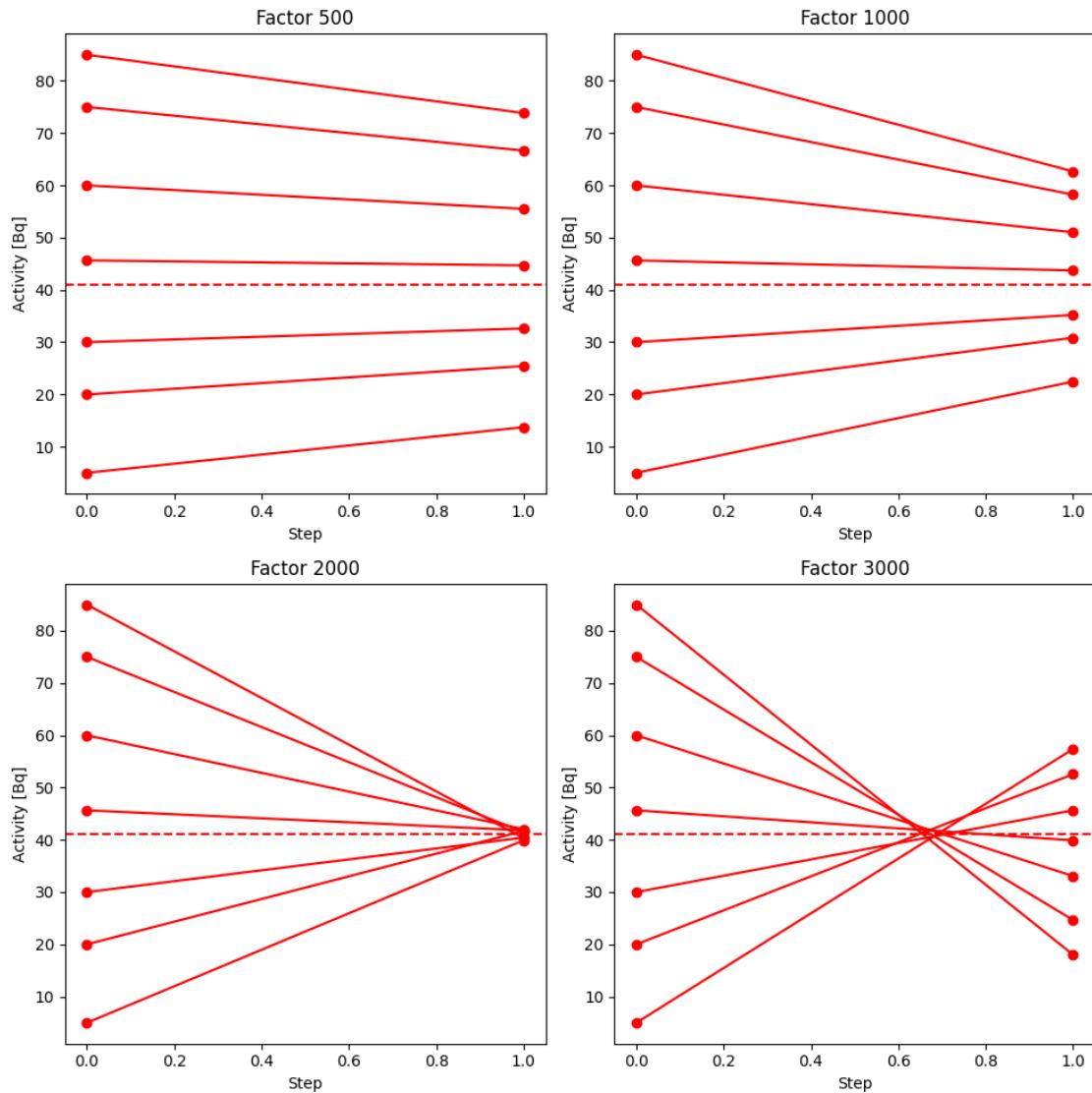
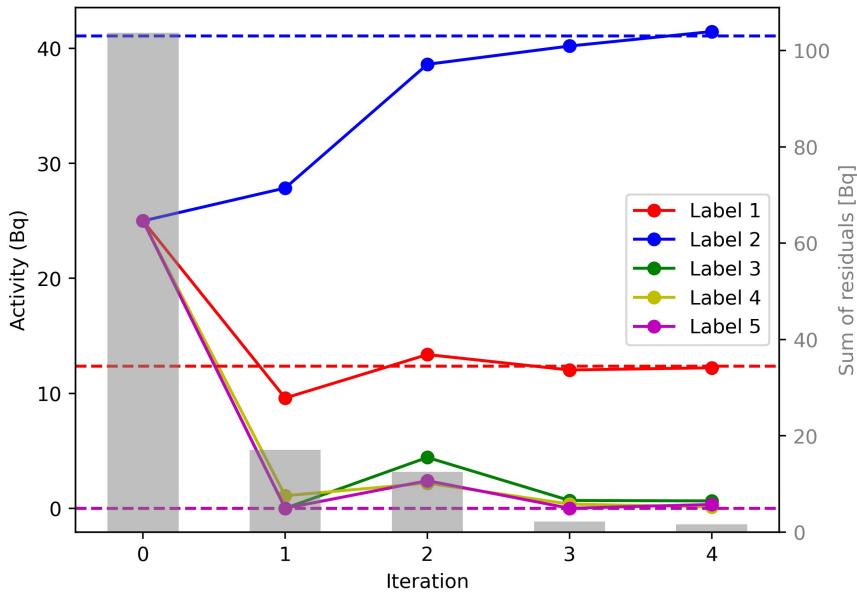


Figure 6.6: Experiments for different starting values with different factors  $f$ . The dashed line represents the ground truth.



*Figure 6.7: Iteration steps with GATE visualized with two active labels. The dashed lines are the ground truth. As Label 3 to 5 are all at 0, only one dash line is visible. The grey bars represent the sum of the residuals between the current activity estimate and the ground truth for all five labels. For initialization, all activities were set to 25Bq.*

noise might be more prominent (see Fig. 6.8). Again, the initialization was set to the same value for all labels. In the first update step, it overshoots quite strongly. This is because Label 1 (thyroid) is by far the biggest label and influences the overall hit rate the most. As this value comes close to its ground truth, the other values quickly correct in the next iteration step to the ground truth. Once the three small Labels 3 to 5 are at their activity range, no major changes are visible. Very low activities together with small labels are very hard to converge for the algorithm. This is because of the stochastic nature of the GATE simulation. Only very few pixels in the image are affected by those labels. Minor stray hits from other labels, such as the thyroid, affect the update calculation strongly. The whole convergence took 32.78 hours excluding the target simulation.

In the third experiment, all five labels are active again with overall high activity (see Fig. 6.9). Initialization for all labels was set to 25 Bq. This experiment is comparable to the previous experiment but with higher overall activities. All five labels converge to the ground truth. Also, the small labels were now set with high activity. With higher activities and differences between other labels next to them (Label 1), they converge much better with less noise to the ground truth. Excluding the target simulation, the five update steps took 42.35 hours.

In the last experiment, the algorithm was tested with many difficulties at the same

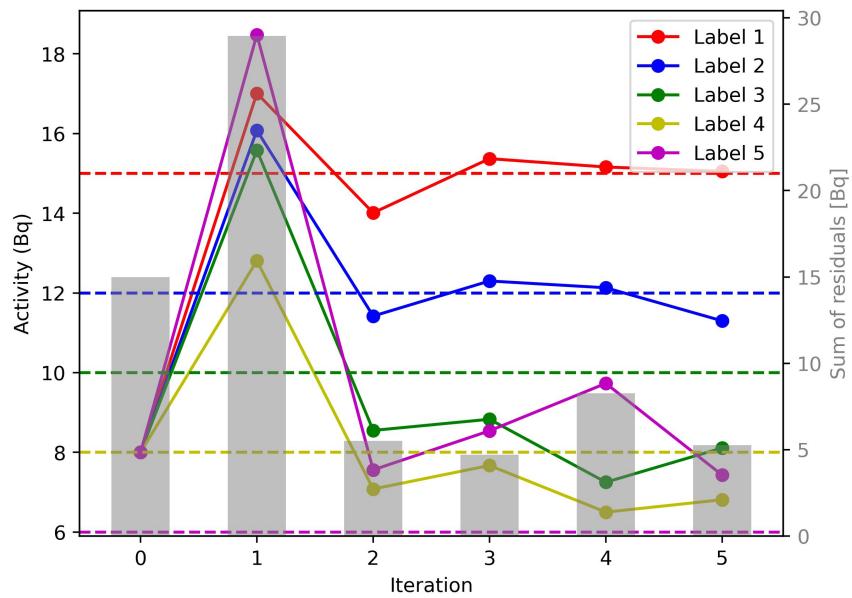


Figure 6.8: Iteration steps with GATE visualized with all labels active but an overall low activity. For initialization, all activities were set to 8 Bq.

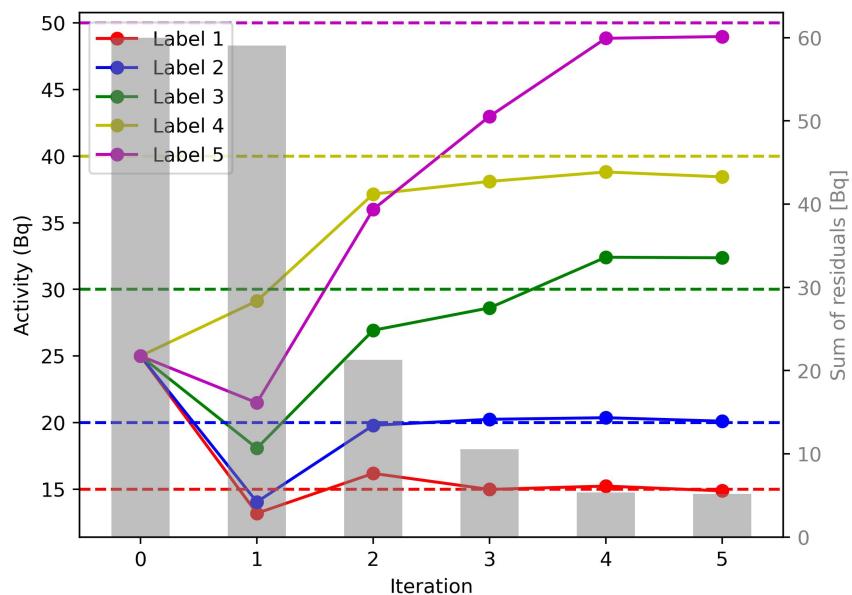


Figure 6.9: Iteration steps with GATE visualized with all labels active and overall high activity. For initialization, all activities were set to 25 Bq.

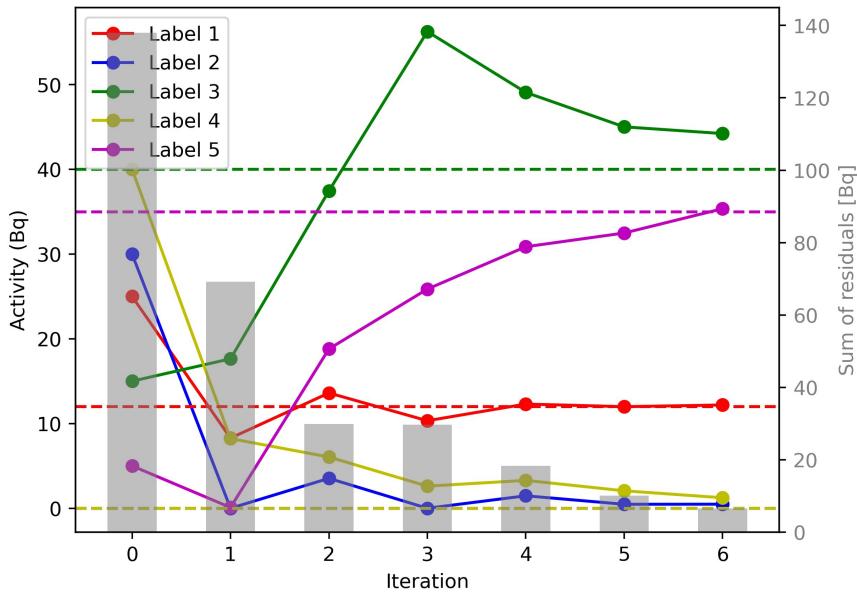


Figure 6.10: Iteration steps with GATE visualized with three labels active and two set to zero. For initialization, the activities were set to 25 Bq, 30 Bq, 15 Bq, 40 Bq, 5 Bq.

time. This means, starting with different initialization values for all five labels, some labels are set to active, while others are set to 0, with high differences in activities ranging from 40 Bq to 0 Bq (see Fig. 6.10). The algorithm solves this task very well, by reducing the SOR in five steps from 138 Bq to 10.07 Bq. It took 40.35 hours to converge.

All experiments demonstrate the robust performance of the algorithm and demonstrate a proof-of-concept for the overall project. In the next section, the GATE simulations are replaced by the rendering combined with a neural network.

#### 6.4.2 Rendering + Neural Net

The same four experiments for the iteration loop with GATE were performed using the renderings and the neural network instead of GATE. Again, GATE-simulated target images were used in the experiments, as they represent the ground truth. This time, all iteration loops were run for six update steps, as time was not a limitation.

The first experiment with the neural network and rendering had a starting value of 25 Bq for each label and only two active target labels. It can be seen that the iteration loop performs very well given the SOR is reduced (see Fig. 6.11). The two active labels are predicted accurately after a few iteration steps. Compared to the 33.41 hours runtime of the GATE-only experiment (with 4 update steps), the

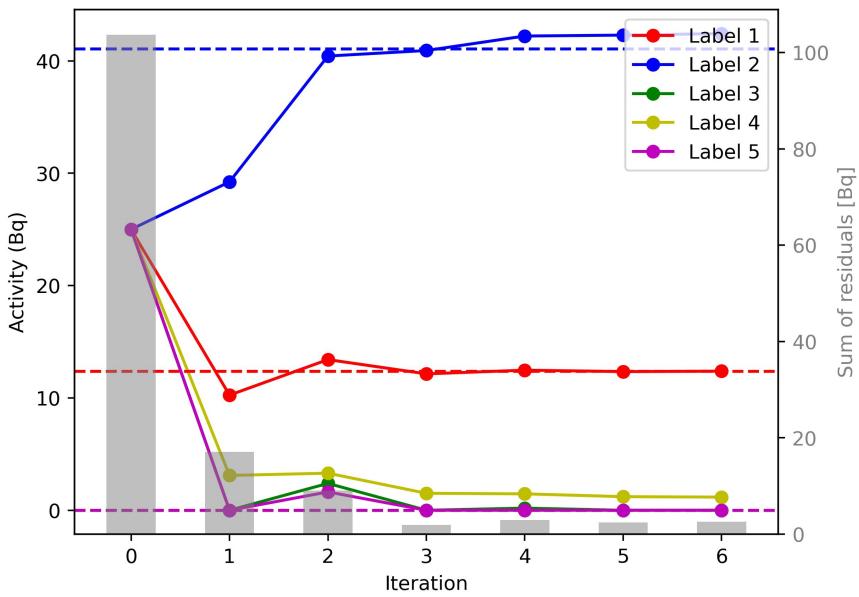


Figure 6.11: Iteration steps with the rendering + neural net visualized with two active labels. The dashed lines are the ground truth. As Label 3 to 5 are all at 0, only one dash line is visible. For initialization, all activities were set to 25Bq. The GATE equivalent can be seen in Fig. 6.7.

rendering and neural net version took only 252 seconds to run 6 update steps. This is only 0.2% of the original time.

In the second experiment, all labels were active but at a very low activity (see Fig. 6.12). After 2 update steps, the iteration does not improve significantly anymore. This is due to the fact that Label 5 is predicted with too high activity. This difference only improved minorly afterward. The reason for this wrong behavior is the neural network having trouble translating the hit rates of these pixels (Label 5) from the rendering to reality/GATE at this activity level. However, all other nodules perform well, especially the two biggest labels Label 1 and Label 2. For the other two low-activity labels, Label 4 converges to its ground truth well while Label 3 is underestimated. This behavior for wrong low activities was seen in the GATE-only versions as well. The whole iteration took 253 seconds to complete. The rendering and neural networks runtime does not depend on the activity, hence it is very similar in all the experiments. The same setup took 32.78 hours to complete with the GATE-only version. This is a time reduction of more than 99% again.

For the third experiment, a different, but high activity was set for all labels. All starting values were set to 25 Bq (see Fig. 6.13). Again, Labels 1 and 2 converge very well to the ground truth, while Label 5 is wrong and diverges from the ground truth after 4 iteration steps. This behavior can potentially be explained by the same

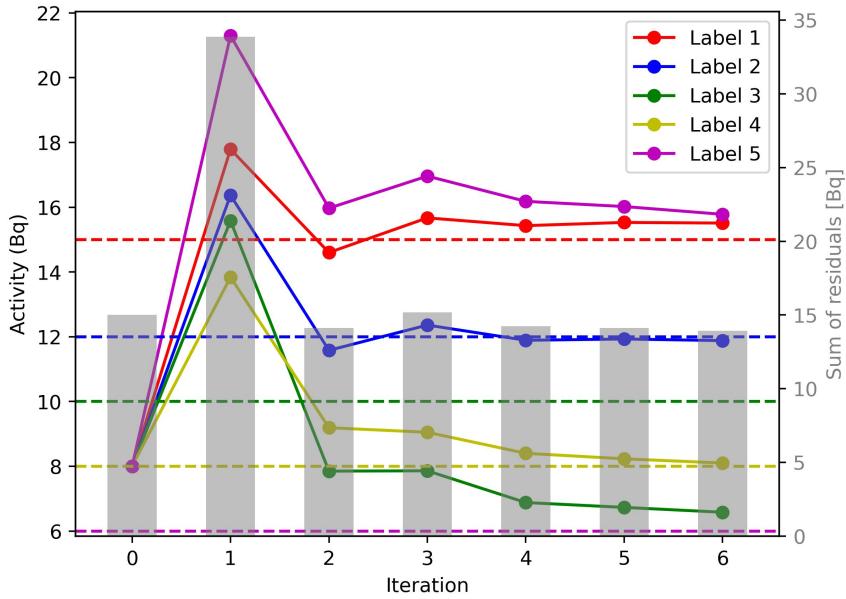


Figure 6.12: Iteration steps with the rendering + neural net visualized with all labels active but an overall low activity. For initialization, all activities were set to 8 Bq. The GATE equivalent can be seen in Fig. 6.8.

reason as in the previous experiment: The neural network seems to assign wrong hit counts to this very small Label 5, which influences the overall SOR and increases it starting from step 3. All 6 steps took 257 seconds to complete compared to 42.35 hours with GATE.

The last experiment started with different initialization values, where some labels were set to active and others to 0 (see Fig. 6.14). The iteration loop converges very well on all labels and SOR reduces over time. Label 5 is predicted at high activity and still increasing after 6 steps. This indicates again that the neural network tends to give this label high activity. As the ground truth of this label was also high, the difference is not as noticeable as in the previous examples. The whole algorithm took 250 seconds to complete.

All experiments show that the iteration loop algorithm performs as expected. The algorithm itself is very robust to many different settings. For low activities in small labels, it has its limitations. But those can potentially be eradicated with a more realistic GATE simulation, for example by simulating for more than ten seconds. Ten seconds are only 3% of the normal 300 seconds of thyroid scintigraphy, making it very sensitive to noise. This noise affects only a few pixels in the image. However, very small labels are only projected to very few pixels in the image. Changes in these few pixels affect the iteration loops' performance significantly. Additionally, differences in a few pixels are minor in the loss and metrics used, making it hard

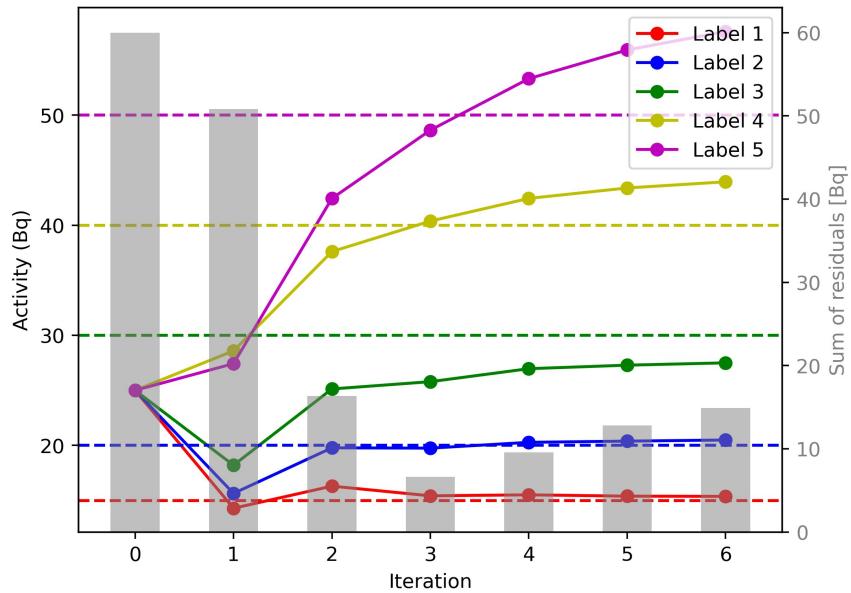


Figure 6.13: Iteration steps with the rendering + neural net visualized with all labels active and overall high activity. For initialization, all activities were set to 25 Bq. The GATE version can be seen in Fig. 6.9.

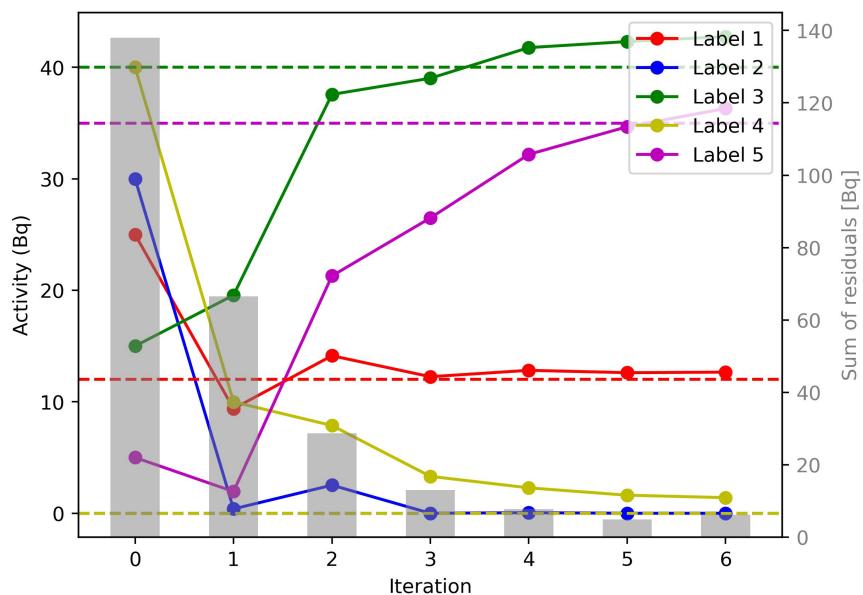


Figure 6.14: Iteration steps with the rendering + neural net visualized with all labels active and overall high activity. For initialization, all activities were set to 25 Bq. The GATE version can be seen in Fig. 6.10.

for the neural network to predict correct values. This reasoning is supported by the fact that in all experiments, the big Labels 1 and 2 are very accurately predicted. Overall, the iteration algorithm works as expected and converges to its ground truth in most cases. When the rendering and neural network are used instead of GATE, the algorithm converges in less than 260 seconds. Limitations occurred but are assumed to be due to the simulations of the scintigraphies rather than the iteration algorithm itself.



# Chapter 7

## Conclusion and Outlook

The aim of this thesis was to develop an algorithm for iteratively estimating the 3D activity uptake in the thyroid and nodules using only a segmented 3D ultrasound (US), the relative pose of the US to the camera, and a 2D scintigraphy. This was achieved by simulating a scintigraphy given a manually defined 3D-activity distribution, which is compared to the real scintigraphy. Based on the difference, the 3D-activity distribution is iteratively updated until the simulated and real scintigraphies are matching. To simulate scintigraphies a Monte-Carlo simulator was implemented and validated against real-world measurements. Due to the clinically unfeasible simulation time, a neural network was trained with the Monte-Carlo simulations as ground truth. This reduced the simulation time by more than 99%, while still achieving similar results. The iteration algorithm was tested on many cases and has shown robust performance.

In future research, the Monte-Carlo simulator can be further improved by including more modules, simulating for a longer time, and validating more diverse real-world measurements against the simulation. Additionally, the discussed relationship between solid-angle and distance to the detector can be investigated. A lack of patient data of matching scintigraphies and labeled 3D-US scans guided us toward the use of a 3D-printed phantom to validate the proof-of-concept. In future research, this algorithm can be extended to use with patient data. However, several steps are necessary to achieve this. First, 3D-US usually consists of two separate US volumes of the right and left side of the beck. To get one volume, the two volumes have to be registered. Additionally, this registered volume needs to be segmented with as many labels as possible, ideally automatically. Especially for the thyroid and nodules inside the thyroid, the segmentation has to be very accurate for the algorithm to produce sufficient results. The neural network translates a rendered projection to a more noisy/realistic one with a U-net architecture. In the literature review diffusion

models were introduced, which promise strong performance for this image-to-image task and can be investigated in future research. Another approach could be investigated by replacing the projection, neural network, and Monte-Carlo simulation with one neural network. This neural network takes 3D labels, the relative position of the labels to a detector, and a reference scintigraphy as input and produces a matching 3D-activity map. Due to the high dimensionality of the data, a high amount of high-quality labeled and matched patient data is necessary to train the neural network. Currently, this dataset is not available but could be generated in the future with clinical studies. Overall, this thesis already demonstrates a proof-of-concept for estimating the 3D activity uptake in the thyroid.

# Bibliography

- [1] S. Agostinelli, J. Allison, K. Amako, J. Apostolakis, H. Araujo, P. Arce, M. Asai, D. Axen, S. Banerjee, G. Barrand, F. Behner, L. Bellagamba, J. Boudreau, L. Broglia, A. Brunengo, H. Burkhardt, S. Chauvie, J. Chuma, R. Chytracek, G. Cooperman, G. Cosmo, P. Degtyarenko, A. Dell'Acqua, G. Depaola, D. Dietrich, R. Enami, A. Feliciello, C. Ferguson, H. Fesefeldt, G. Folger, F. Foppiano, A. Forti, S. Garelli, S. Giani, R. Giannitrapani, D. Gibin, J.J. Gómez Cadenas, I. González, G. Gracia Abril, G. Greeniaus, W. Greiner, V. Grichine, A. Grossheim, S. Guatelli, P. Gumplinger, R. Hamatsu, K. Hashimoto, H. Hasui, A. Heikkinen, A. Howard, V. Ivanchenko, A. Johnson, F.W. Jones, J. Kallenbach, N. Kanaya, M. Kawabata, Y. Kawabata, M. Kawaguti, S. Kelner, P. Kent, A. Kimura, T. Kodama, R. Kokoulin, M. Kossov, H. Kurashige, E. Lamanna, T. Lampén, V. Lara, V. Lefebure, F. Lei, M. Liendl, W. Lockman, F. Longo, S. Magni, M. Maire, E. Medernach, K. Minamimoto, P. Mora de Freitas, Y. Morita, K. Murakami, M. Nagamatu, R. Nartallo, P. Nieminen, T. Nishimura, K. Ohtsubo, M. Okamura, S. O'Neale, Y. Oohata, K. Paech, J. Perl, A. Pfeiffer, M.G. Pia, F. Ranjard, A. Rybin, S. Sadilov, E. Di Salvo, G. Santin, T. Sasaki, N. Savvas, Y. Sawada, S. Scherer, S. Sei, V. Sirotenko, D. Smith, N. Starkov, H. Stoecker, J. Sulkimo, M. Takahata, S. Tanaka, E. Tcherniaev, E. Safai Tehrani, M. Tropeano, P. Truscott, H. Uno, L. Urban, P. Urban, M. Verderi, A. Walkden, W. Wander, H. Weber, J.P. Wellisch, T. Wenaus, D.C. Williams, D. Wright, T. Yamada, H. Yoshida, and D. Zschiesche. Geant4—a simulation toolkit. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 506(3):250–303, 2003.
- [2] Awad Alshahrani, Gulam Syed, Altaf Khan, Shah Numani, Abdulaziz Alnaim, Faisal Alanazi, and Moeber Mahzari. Assessment of normal reference values for thyroid uptake of technetium-99m pertechnetate in a saudi population. *Annals of Saudi Medicine*, 41:86–90, 04 2021.

- [3] Michael W. Ashcraft and Andre J. Van Herle. Management of thyroid nodules. i: History and physical examination, blood tests, x-ray tests, and ultrasonography. *Head & Neck Surgery*, 3(3):216–227, 1981.
- [4] Michael W. Ashcraft and Andre J. Van Herle. Management of thyroid nodules. ii: Scanning techniques, thyroid suppressive therapy, and fine needle aspiration. *Head & Neck Surgery*, 3(4):297–322, 1981.
- [5] Harold L. Atkins. Technetium-99m pertechnetate uptake and scanning in the evaluation of thyroid function. *Seminars in nuclear medicine*, 1 3:345–55, 1971.
- [6] Vertes Attila, Nagy Sandor, Klencsar Zoltan, Lovas Rezso G., and Rosch Frank. *Handbook of Nuclear Chemistry: Basics of Nuclear Science; Elements and Isotopes: Formation, Transformation, Distribution; Chemical Applications of Nuclear Reactions and Radiations; Radiochemistry and Radiopharmaceutical Chemistry in Life Sciences; Instrumentation, Separation Techniques, Environmental Issues*, pages 1267–1297. Springer US, Boston, MA, 2004.
- [7] I K Bailey, L S Griffith, J Rouleau, W Strauss, and B Pitt. Thallium-201 myocardial perfusion imaging at rest and during exercise. comparative sensitivity to electrocardiography in coronary artery disease. *Circulation*, 55(1):79–87, 1977.
- [8] D Becker, N D Charles, H Dworkin, J Hurley, I R McDougall, D Price, H Royal, and S Sarkar. Procedure guideline for thyroid scintigraphy: 1.0. *Journal of nuclear medicine : official publication, Society of Nuclear Medicine*, 7(37):1264–6, 1996.
- [9] Lukas Biewald. Experiment tracking with weights and biases, 2020. Software available from wandb.com.
- [10] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020.
- [11] Marie-Martine Bé, V. Chisté, Christophe Dulieu, Edgardo Browne, V. Chechev, Nikolay Kuzmenko, Richard Helmer, Alan Nichols, Eckart Schönfeld, and

- Rainer Dersch. *Table of radionuclides (Vol. 1 - A = 1 to 150)*. Bureau International des Poids et Mesures - BIPM, 01 2004.
- [12] M. Jorge Cardoso, Wenqi Li, Richard Brown, Nic Ma, Eric Kerfoot, Yiheng Wang, Benjamin Murray, Andriy Myronenko, Can Zhao, Dong Yang, Vishwesh Nath, Yufan He, Ziyue Xu, Ali Hatamizadeh, Wentao Zhu, Yun Liu, Mingxin Zheng, Yucheng Tang, Isaac Yang, Michael Zephyr, Behrooz Hashemian, Sachidanand Alle, Mohammad Zalbagi Darestani, Charlie Budd, Marc Modat, Tom Vercauteren, Guotai Wang, Yiwen Li, Yipeng Hu, Yunguan Fu, Benjamin Gorman, Hans Johnson, Brad Genereaux, Barbaros S. Erdal, Vikash Gupta, Andres Diaz-Pinto, Andre Dourson, Lena Maier-Hein, Paul F. Jaeger, Michael Baumgartner, Jayashree Kalpathy-Cramer, Mona Flores, Justin Kirby, Lee A.D. Cooper, Holger R. Roth, Daguang Xu, David Bericat, Ralf Floca, S. Kevin Zhou, Haris Shuaib, Keyvan Farahani, Klaus H. Maier-Hein, Stephen Aylward, Prerna Dogra, Sebastien Ourselin, and Andrew Feng. MONAI: An open-source framework for deep learning in healthcare. November 2022.
- [13] Vincent W. S. Chan and Sherif Abbas. *Ultrasound imaging for regional anesthesia : a practical guide*. Ultrasound for Regional Anesthesia, Toronto, 2009.
- [14] Simon R. Cherry, James A. Sorenson, and Michael E. Phelps. *Physics in nuclear medicine*. Elsevier Health Sciences, 2012.
- [15] Cleveland Clinic. Thyroid nodules, 2022. URL = <https://my.clevelandclinic.org/health/diseases/13121-thyroid-nodule> accessed 2022-09-22.
- [16] David. S. Cooper. Revised american thyroid association management guidelines for patients with thyroid nodules and differentiated thyroid cancer. *Thyroid*, 19(11):1167–214, 2009.
- [17] Christian David, Zhihua Qi, and Guang-Hong Chen. Direct fan-beam reconstruction algorithm via filtered backprojection for differential phase-contrast computed tomography. *X-Ray Optics and Instrumentation*, 2008:835172, 2008.
- [18] Wylie J. Dodds and Malcolm R. Powell. Thyroid scanning with technetium 99m pertechnetate. *Radiology*, 91(1):27–31, 1968. PMID: 5654032.
- [19] Ulrich Eck. Cpu-based volume visualization (camp2 lecture exercise). personal communication.

- [20] Frederic H Fahey, Kira Grogg, and Georges El Fakhri. Use of monte carlo techniques in nuclear medicine. *J. Am. Coll. Radiol.*, 15(3 Pt A):446–448, March 2018.
- [21] William Falcon and The PyTorch Lightning team. PyTorch Lightning, March 2019. <https://github.com/Lightning-AI/lightning>.
- [22] Nuclear Fields. Collimators for nuclear medicine, 2023. URL = <https://www.nuclearfields.com/collimators-nuclear-medicine.htm> accessed 2023-03-15.
- [23] Geant4. Em opt4, 2022. url = <https://geant4-userdoc.web.cern.ch/UsersGuides/PhysicsListGuide/html/electromagnetic/Opt4.html>.
- [24] AW Goolden, HI Glass, and ED Williams. Use of  $^{99}\text{tc m}$  for the routine assessment of thyroid function. *British medical journal*, 4(5784):396—399, November 1971.
- [25] John E. Hall and Arthur C. Guyton. *Guyton and Hall textbook of medical physiology*, pages 907–910. Saunders Elsevier, Philadelphia, PA, twelfth edition, 2011.
- [26] Zhihui Han, Hu Peng, and Jingwen Pan. A two-steps implementation of 3d ultrasound imaging in frequency domain with 1d array transducer. *Ultrasonics*, 114:106423, 2021.
- [27] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2 edition, 2004.
- [28] Laszlo Hegedüs. Clinical practice. the thyroid nodule. *The New England journal of medicine*, 351(17):1764–1771, 2004.
- [29] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS’20, Red Hook, NY, USA, 2020. Curran Associates Inc.
- [30] AMERICAN NATIONAL STANDARDS INSTITUTE. Ansi, 2022. URL = <https://www.ansi.org/> accessed 2022-09-29.
- [31] S Jan, G Santin, D Strul, S Staelens, K Assié, D Autret, S Avner, R Barbier, M Bardies, P M Bloomfield, D Brasse, V Breton, P Bruyndonckx, I Buvat, A F Chatzioannou, Y Choi, Y H Chung, C Comtat, D Donnarieix, L Ferrer,

- S J Glick, C J Groiselle, D Guez, P F Honore, S Kerhoas-Cavata, A S Kirov, V Kohli, M Koole, M Krieguer, D J van der Laan, F Lamare, G Largeron, C Lartizien, D Lazaro, M C Maas, L Maigne, F Mayet, F Melot, C Merheb, E Pennacchio, J Perez, U Pietrzyk, F R Rannou, M Rey, D R Schaart, C R Schmidlein, L Simon, T Y Song, J M Vieira, D Visvikis, R Van de Walle, E Wieërs, and C Morel. GATE: a simulation toolkit for PET and SPECT. *Phys. Med. Biol.*, 49(19):4543–4561, October 2004.
- [32] Sama Jan, D Benoit, E. Becheva, Thomas Carlier, Franca Cassol, Patrice Descourt, Thibault Frisson, Loïc Grevillot, Laurent Guigues, Lydia Maigne, Christian Morel, Yann Perrot, N Rehfeld, David Sarrut, Dennis Schaart, S Stute, Uwe Pietrzyk, D Visvikis, Nabil Zahra, and Irene Buvat. Gate v6: A major enhancement of the gate simulation platform enabling modelling of ct and radiotherapy. *Physics in Medicine and Biology*, 56:881, 01 2011.
- [33] Han Gyu Kang. Gate training on medical imaging (pet, spect, ct), dosimetry and radiation therapy – beginner level, session8, 2020. accessed 2022-05-24.
- [34] Mingyu Kim, Jihye Yun, Yongwon Cho, Keewon Shin, Ryoungwoo Jang, Hyun-Jin Bae, and Namkug Kim. Deep learning in medical imaging. *Neurospine*, 16(4):657–668, December 2019.
- [35] T Klein, Mattias Hansson, and Nassir Navab. Modeling of multi-view 3d freehand radio frequency ultrasound. *Medical image computing and computer-assisted intervention : MICCAI ... International Conference on Medical Image Computing and Computer-Assisted Intervention*, 15:422–9, 10 2012.
- [36] Rüdiger Kramme. *Medizintechnik*. Springer, Berlin,Heidelberg, 2011.
- [37] Z Kusić, DV Becker, EL Saenger, P Paras, P Gartside, T Wessler, and S Spaventi. Comparison of technetium-99m and iodine-123 imaging of thyroid nodules: correlation with pathologic findings. *Journal of nuclear medicine : official publication, Society of Nuclear Medicine*, 31(4):393—399, April 1990.
- [38] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.
- [39] Brian Lentz, Tiffany Fong, Randall Rhyne, and Nicholas Risko. A systematic review of the cost-effectiveness of ultrasound in emergency care settings. *The Ultrasound Journal*, 13(1):16, 2021.

- [40] Mathias Lukas, Anne Kluge, Nicola Beendorff, and Winfried Brenner. Accurate monte carlo modeling of small-animal multi-pinhole spect for non-standard multi-isotope applications. *IEEE Transactions on Medical Imaging*, PP:1–1, 04 2021.
- [41] EL Mazzaferri. Management of a solitary thyroid nodule. *The New England journal of medicine*, 328(8):553—559, February 1993.
- [42] Johannes Meller and W Becker. The continuing importance of thyroid scintigraphy in the era of high-resolution ultrasound. *European journal of nuclear medicine and molecular imaging*, 29 Suppl 2:S425–38, 09 2002.
- [43] Mehdi Momennezhad, Ramin Sadeghi, and Shahrokh Nasseri. Development of gate monte carlo simulation for a dual-head gamma camera. *Radiological Physics and Technology*, 5(2):222–228, 2012.
- [44] Shorook Na’ara, Kamel Mahameed, Moran Amit, Jacob T. Cohen, Michal Weiler-Sagie, Igor Albitskiy, Ziv Gil, and Salem Billan. Efficacy of posttreatment radioiodine scanning in patients with differentiated thyroid cancer. *Head & Neck*, 41(9):3235–3240.
- [45] Matthias Nießner and Laura Leal-Taixé. Introduction to deep learning lecture. personal communication, 2021. URL = [https://www.youtube.com/watch?v=Xa8d\\_j5f2pI&list=PLQ8Y4kIIbzy\\_0aXv861fbQwPHSomk2o2e&index=9&ab\\_channel=MatthiasNiessner](https://www.youtube.com/watch?v=Xa8d_j5f2pI&list=PLQ8Y4kIIbzy_0aXv861fbQwPHSomk2o2e&index=9&ab_channel=MatthiasNiessner) accessed 2023-03-14.
- [46] OpenAI. Chatgpt, 2022. [Computer software], url = <https://openai.com/blog/chatgpt>.
- [47] Catherine M. Otto. *Textbook of clinical echocardiography*. Elsevier, Philadelphia, 2004.
- [48] Franz Pfeiffer. Lecture biomedical physics 1. personal communication.
- [49] Geanina Popoveniuc and Jacqueline Jonklaas. Thyroid nodules. *The Medical clinics of North America*, 96(2):329–49, 2012.
- [50] Stephen P. Povoski, Douglas A. Murray, Sabrina M. Smith, Edward W. Martin, and Nathan C. Hall. 18f-fdg pet/ct oncologic imaging at extended injection-to-scan acquisition time intervals derived from a single-institution 18f-fdg-directed surgery experience: feasibility and quantification of 18f-fdg accumulation within 18f-fdg-avid lesions and background tissues. *BMC Cancer*, 14(1):453, 2014.

- [51] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *ArXiv*, abs/2204.06125, 2022.
- [52] Robert Rohling, Andrew Gee, and Laurence Berman. A comparison of free-hand three-dimensional ultrasound reconstruction techniques. *Medical Image Analysis*, 3(4), 1999.
- [53] Maria T. Rojeski and Hossein Gharib. Nodular thyroid disease. *New England Journal of Medicine*, 313(7):428–436, 1985. PMID: 3894966.
- [54] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.
- [55] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham, 2015. Springer International Publishing.
- [56] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986.
- [57] Chitwan Saharia, William Chan, Huiwen Chang, Chris A Lee, Jonathan Ho, Tim Salimans, David J Fleet, and Mohammad Norouzi. Palette: Image-to-Image diffusion models. *arXiv*, 2021.
- [58] Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J. Fleet, and Mohammad Norouzi. Image super-resolution via iterative refinement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(4):4713–4726, 2023.
- [59] Sarrut, David and Bała, Mateusz and Bardies, Manuel and Bert, Julien and Chauvin, Maxime and Chatzipapas, Konstantinos and Dupont, Mathieu and Etxebeste, Ane and Fanchon, Louise M and Jan, Sébastien and Kayal, Gunjan and Kirov, Assen S and Kowalski, Paweł and Krzemien, Wojciech and Labour, Joey and Lenz, Mirjam and Loudos, George and Mehadji, Brahim and Ménard, Laurent and Morel, Christian and Papadimitroulas, Panagiotis and Rafecas, Magdalena and Salvadori, Julien and Seiter, Daniel and Stockhoff, Mariele and

- Testa, Etienne and Trigila, Carlotta and Pietrzyk, Uwe and Vandenberghe, Stefaan and Verdier, Marc-Antoine and Visvikis, Dimitris and Ziemons, Karl and Zvolský, Milan and Roncali, Emilie. Advanced Monte Carlo simulations of emission tomography imaging systems with GATE. *PHYSICS IN MEDICINE AND BIOLOGY*, 66(10):23, 2021.
- [60] Wolfgang Schlegel, Christian P. Karger, and Oliver Jäkel. *Medizinische Physik*. Springer Spektrum, Berlin, Heidelberg, 2018.
- [61] YU.A. SHREIDER, N.P. BUSLENKO, D.I. GOLENKO, I.M. SOBOL', and V.G. SRAGOVICH. Chapter i - principles of the monte carlo method. In *The Monte Carlo Method*, pages 1–90. Pergamon, 1966.
- [62] Sean Smith, W. Lee, E.D. Light, J.T. Yen, Patrick Wolf, and S. Idriss. Two dimensional arrays for 3-d ultrasound imaging. *Proceedings of the IEEE Ultrasonics Symposium*, 2:1545 – 1553 vol.2, 11 2002.
- [63] Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, page 2256–2265. JMLR.org, 2015.
- [64] Steven Staelens, Tim de Wit, Ignace Lemahieu, and Freek Beekman. Degradation of myocardial perfusion spect images caused by contaminants in thallous (tl-201) chloride. *European journal of nuclear medicine and molecular imaging*, 35:922–32, 06 2008.
- [65] Matthew Evan Strugari, Drew Roderick DeBay, Steven Donald Beyea, and Kimberly Dawn Brewer. Nema nu 1-2018 performance characterization and monte carlo model validation of the cubresa spark sipm-based preclinical spect scanner, 2022.
- [66] G H Tan and H Gharib. Thyroid incidentalomas: Management approaches to nonpalpable nodules discovered incidentally on thyroid imaging. *Annals of Internal Medicine*, 126(3):226–231, 1997. PMID: 9027275.
- [67] Karen Van Audenhaege, Roel Van Holen, Stefaan Vandenberghe, Christian Vanhove, Scott Metzler, and Stephen Moore. Review of spect collimator selection, optimization, and fabrication for clinical and preclinical imaging. *Medical physics*, 42:4796, 08 2015.

- [68] Mark P J Vanderpump. The epidemiology of thyroid disease. *Br. Med. Bull.*, 99(1):39–51, 2011.
- [69] Sandra Vieira, Walter Pinaya, and Andrea Mechelli. Using deep learning to investigate the neuroimaging correlates of psychiatric and neurological disorders: Methods and applications. *Neuroscience & Biobehavioral Reviews*, 74, 01 2017.
- [70] Tonghe Wang, Yang Lei, Yabo Fu, Jacob F Wynne, Walter J Curran, Tian Liu, and Xiaofeng Yang. A review on medical imaging synthesis using deep learning and its clinical applications. *J. Appl. Clin. Med. Phys.*, 22(1):11–36, January 2021.
- [71] Justus Weber, Uwe Haberkorn, and Walter Mier. Cancer stratification by molecular imaging. *International journal of molecular sciences*, 16:4918–4946, 03 2015.
- [72] Wikipedia. Frequency, 2022. URL = <https://en.wikipedia.org/wiki/Frequency> accessed 2022-09-29.
- [73] Wikipedia. Zum raumwinkel einer pyramidey, 2022. URL = [https://de.wikipedia.org/wiki/Raumwinkel#/media/Datei:SolidAngle\\_Pyramid.png](https://de.wikipedia.org/wiki/Raumwinkel#/media/Datei:SolidAngle_Pyramid.png) accessed 2023-03-20.
- [74] H O Wyckoff, A Allisy, and K Lidén. The new special names of SI units in the field of ionizing radiations. *Radiology*, 118(1):233–234, January 1976.
- [75] zacharmarz. most efficient aabb vs ray collision algorithms, 2011. URL = <https://gamedev.stackexchange.com/questions/18436/most-efficient-aabb-vs-ray-collision-algorithms> accessed 2023-03-20.
- [76] Yang Zhou, Sudipta Chakraborty, and Shuang Liu. Radiolabeled cyclic rgd peptides as radiotracers for imaging tumors and thrombosis by spect. *Theranostics*, 1:58–82, 01 2011.
- [77] Sibylle Ziegler. Nuclear imaging principles and technology in nuclear medicine (camp1 lecture). personal communication.



# List of Figures

2.1	Illustration of a thyroid and thyroid nodules. [15]	4
2.2	Patient with myxedema. (Courtesy Dr. Herbert Langford.) [25]	5
2.3	In radiology the source is outside of the patient (left) while in nuclear medicine the source is within the patient (right). [77]	6
2.4	Electromagnetic spectrum with different regions. Visible light (Vis), Ultraviolet light (UV). [14]	7
2.5	Schematic illustration of a gamma camera with the components: parallel collimator, scintillation crystal, PMTs, and readout electronics.	7
2.6	Schematic illustration of a photomultiplier tube. [14]	8
2.7	Thyroid scintigraphy of a patient. The dark background and edge are due to the shielding around the collimator. In this area, no hits arrive at the crystal.	10
2.8	Spectrum of sound with annotations of specific regions [72].	11
2.9	Penetration depth and wavelengths for different frequencies [47].	12
2.10	Principle of A-mode image generation in ultrasound. [48]	13
2.11	Ultrasound image of the left side of the neck inside a patient with the thyroid visible.	14
2.12	Principle how to generate a 3D volume from tracked US images [52][35].	14
2.13	Illustration of the Fourier slice theorem in reconstruction.	16
2.14	Steps involved in the filtered back projection algorithm.	16
2.15	The figure illustrates the constituent elements of a neural network in (a). A unit is depicted that accepts inputs $x$ and aggregates them by means of weighted summation utilizing the weights $w$ . Following this, an activation function $f$ is employed to introduce non-linearities, thereby generating the output $y$ , which subsequently serves as the input for the next layer (b). The hyperparameters of the neural network, such as the number of layers, the type of activation function, and the interconnections, impact the network's efficacy significantly. This figure has been sourced from [69].	19

2.16 Example of a 2D convolution using a $3 \times 3$ filter on a $5 \times 5$ image as input. The output is a $3 \times 3$ image. In this operation, only 9 weights are necessary, as the kernel has size $3 \times 3$ [45]. . . . .	20
2.17 U-net architecture: blue boxes mean a multi-channel feature map. The number of channels is denoted on top of each box and the x-y-size is at the lower left edge. White boxes represent copied feature maps. . . . .	21
2.18 Results of Palette: Image-to-Image translation tasks with examples of colorization, inpainting, uncropping, and JPEG restoration [57]. . .	23
2.19 Architecture of the latent diffusion model with conditioning via concatenation or a general cross-attention mechanism [54]. . . . .	24
3.1 Overview of the proposed algorithm to estimate a 3D voxelized activity map: Step 1: A 3D material map and an initially guessed activity map are created based on a segmented 3D-US. Step 2: The 3D activity map, the material map, and the relative pose of the US to the camera are used to create an orthographic projection. Step 3: The orthographic projection is the input for a style transfer neural network, yielding a simulated scintigraphy. Step 4: The simulated scintigraphy is compared to the real scintigraphy. Step 5: Based on the difference, the change per label is calculated. Step 6: This change is added to the initially guessed 3D activity map. Step 7: Repeat Step 2 - Step 6 until convergence. . . . .	26
3.2 Left: Highly reflective passive infrared (IR) marker. Right: The dual camera + IR sources . . . . .	28
3.3 A group of markers combined into a target to get the position and orientation of the target. The coordinate system has to be defined when registering the target. . . . .	30
3.4 Example of a camera and a CT-phantom implemented and visualized in GATE. . . . .	31
3.5 A hole of air and two repeaters are used to generate a hexagonal parallel hole collimator. The figure is adapted from [33]. . . . .	31
3.6 The whole camera as a 3D model in GATE with the crystal in red, the cover in green, and the collimator holes in blue. The lead block is drawn as a wireframe to see inside. . . . .	32
3.7 Both extrinsic (a) and intrinsic (b) parameters are visualized [19]. . .	35
3.8 Sketch of the raycasting principle. Rays are depicted in red with $r_n$ and the camera is $T_C$ [19]. . . . .	35

4.1	Sketch and dimensions of the frame model. . . . .	38
4.2	The process of 3d printing. Figure (a) shows the supporting structures as the infill, (b) shows the printing itself and (c) depicts the final product without the tracking target attached. . . . .	39
4.3	The process of 3D printing of the mounts. Figure (a) shows the sketches with dimensions, (b) shows 3D model of the plate and (c) depicts the final product with the tracking target attached but without the phantom, which would be on the other side. . . . .	39
4.4	The 3D models of the phantom. Figure (a) shows the top-down view with dimensions, (b) shows 3D model of the phantom and (c) shows a cross section with the 4 nodules visible inside the thyroid. . . . .	40
4.5	The 3D printing of the phantom. Figure (a) shows the 3D printing with the supporting structure slightly visible, (b) shows the open up-ed phantom and (c) shows the final product printed with a ruler for reference. Also the holes are visible to inject liquids. . . . .	41
4.6	All 3D printed parts including the frame with the tracking target attached, the mounting plate for the phantom with tracking target attached, and two phantoms. . . . .	41
4.7	The dimensions of the targets and their positions are used in the transformations. . . . .	42
4.8	The graph visualizes the tracking with its transformations. Dashed arrows refer to dynamic relations whereas solid arrows mean this transformation is fixed and does not change. To get the unknown transformation, one can walk on the trace of known transformations and concatenate them on the way. This is visualized with the red arrows. . . . .	43
4.9	Coronal slice of a CT of the 3D-printed phantom filled with iodine for higher contrast. This was done to test if there are any leaks in the 3D-printing of the phantom. We did not find any leaks. The lines left and right of the phantom is the bed of the CT device. . . . .	44
4.10	The final CT of the phantom used for attenuation calculations and labelling. Coronal, axial and sagittal planes are visualized together with a 3D rendering. . . . .	45
4.11	The whole setup of the phantom measurements on the gamma camera. . . . .	46
4.12	The three measurements taken with the gamma camera. The pixel values represent the number of hits detected. Parameters for each measurement are summarized in table 4.1. . . . .	47

4.13 1ml syringe used for scintigraphies. . . . .	48
4.14 The three measurements taken with the gamma camera of the syringes. . . . .	48
4.15 Histogramms of the parameters in the dataset. The angles are all displayed in one histogram, even though they are from the different rotation axis. . . . .	49
5.1 Visualizations of the phantom geometry. . . . .	53
5.2 The whole digitizer chain visualized. . . . .	54
5.3 Solid angle of a pyramid [73]. . . . .	57
5.4 Hyperparameter tuning of the model using sweeps. Hyperparameters are the activation function <i>act</i> , the batch size <i>batch_size</i> , depth of the U-net <i>channels</i> , <i>dropout</i> , the learning rate <i>learning_rate</i> , and the loss function. The goal was to minimize the Relative Pixel Sum Loss (RPSL). . . . .	60
5.5 Algorithm to calculate L masks for the iteration loop. Step 1: Project L labels on the imaging plane. Step 2: Expand projection into L channels where each channel represents the count number of this label on the ray. Step 3: Normalize along the channels to get relative percentages of the labels. These masks have to be calculated once at the beginning of the iteration loop. Afterward, they can be reused in each iteration step. . . . .	62
5.6 Algorithm to calculate the change per voxel per label, given the masks. Step 1: The mask for each label is multiplied with the reference and current image, yielding L new images for the current and the reference image. Step 2: All pixel values in each yielded image are summed up and put in a list, yielding a list with L elements for the reference and the current image. Step 3: The difference between these two lists is calculated element-wise and divided by the amount of voxels the initial 3D-Labelmap has for this label. Additionally, all elements are multiplied by a proportionality factor <i>f</i> . . . . .	63
6.1 The three measurements of the phantom together with its reproduced simulation in GATE. In the left column are the measurements of the three phantoms. The right column entails its corresponding simulation in GATE . . . . .	66
6.2 Hit counts of GATE simulations with different distances and sources. . . . .	70

6.3	The rendering (b) compared to the GATE simulations (a) depicted for one sample. The difference of the binary masks of the two images is visualized in (c) and the difference between the raw images is visualized in (d). . . . .	72
6.4	Example of a rendering where the phantom was rotated so that one side is closer to the detector. The parts closer to the detector have a higher count rate as the solid angle is bigger. . . . .	73
6.5	GATE simulation (a), rendering (b), and the output of the neural network (c) made with the same parameters. . . . .	74
6.6	Experiments for different starting values with different factors $f$ . The dashed line represents the ground truth. . . . .	76
6.7	Iteration steps with GATE visualized with two active labels. The dashed lines are the ground truth. As Label 3 to 5 are all at 0, only one dash line is visible. The grey bars represent the sum of the residuals between the current activity estimate and the ground truth for all five labels. For initialization, all activities were set to 25Bq. . .	77
6.8	Iteration steps with GATE visualized with all labels active but an overall low activity. For initialization, all activities were set to 8 Bq. .	78
6.9	Iteration steps with GATE visualized with all labels active and overall high activity. For initialization, all activities were set to 25 Bq. . . .	78
6.10	Iteration steps with GATE visualized with three labels active and two set to zero. For initialization, the activities were set to 25 Bq, 30 Bq, 15 Bq, 40 Bq, 5 Bq. . . . .	79
6.11	Iteration steps with the rendering + neural net visualized with two active labels. The dashed lines are the ground truth. As Label 3 to 5 are all at 0, only one dash line is visible. For initialization, all activities were set to 25Bq. The GATE equivalent can be seen in Fig. 6.7. . . . .	80
6.12	Iteration steps with the rendering + neural net visualized with all labels active but an overall low activity. For initialization, all activities were set to 8 Bq. The GATE equivalent can be seen in Fig. 6.8. . . .	81
6.13	Iteration steps with the rendering + neural net visualized with all labels active and overall high activity. For initialization, all activities were set to 25 Bq. The GATE version can be seen in Fig. 6.9. . . .	82
6.14	Iteration steps with the rendering + neural net visualized with all labels active and overall high activity. For initialization, all activities were set to 25 Bq. The GATE version can be seen in Fig. 6.10. . . .	82



# List of Tables

2.1	Radionuclides together with their half-life, emission energy, and clinical use. [76] [64] [71]	6
2.2	$^{99m}\text{Tc}$ -pertechnetate uptakes in % after 30min in n=1000 adults with euthyroid (healthy), hyperthyroidism and hypothyroidism [5].	10
2.3	Acoustic Impedance values for different tissue types [13].	13
4.1	Parameters of the three phantom measurements. $T_x, T_y$ , and $T_z$ are the translation and $R_x, R_y$ , and $R_z$ the Euler rotation angles from the tracking software.	46
5.1	Gamma emission spectrum of $^{99m}\text{Tc}$ [11].	55
5.2	Hyperparameters used in the final model.	61
6.1	Summary of the total number of hits in the image. The relative difference is calculated via $\frac{\text{GATE}-\text{Measurement}}{\text{Measurement}} \cdot 100$ .	65
6.2	Summary of hit rates for different source types.	67
6.3	Summary of hit rates for different pile-up and dead time combinations.	68
6.4	Parameters used in the GATE simulation and the input for the rendering script. Activities are given per voxel.	71
6.5	Evaluation metrics results on the test set of the final model. The arrows depict if the metric should be small or high. $\uparrow^1$ means that the metric aims to be 1.	73
6.6	Comparison of GATE, rendering, and the neural network with respect to RPSL. An improvement in the RPSL of the neural network is clearly visible. As the GATE image is the reference, it is matching perfectly by definition.	74



# Acknowledgements

Thanks to all people helping and motivating me during this thesis.

First, I want to thank my supervisors Dr. Thomas Wendler, Francesca De Benetti, and Christine Eilers for the excellent supervision and proofreading of the thesis.

I also want to thank the PhDs of the chair for helping me with technical difficulties on the cluster, and the MTRAs in the clinic for helping me with the radioactive measurements.

A big Thank You also goes to all my friends and family for supporting and motivating me throughout my whole studies so far.