UNIVERSITY OF MANNHEIM

MASTER THESIS

Quasi-Monte Carlo Methods and Neural Networks

Author:
Janik V. HRUBANT

Supervisors:
Prof. Dr. Andreas NEUENKIRCH

A thesis submitted in fulfillment of the requirements for the degree of Master of Science

in the

Research Group Name
Department or School Name

July 20, 2025

Declaration of Authorship

I, Janik V. HRUBANT, declare that this thesis titled, "Quasi-Monte Carlo Methods and Neural Networks" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date:

Abstract

This thesis explores the application of quasi-Monte Carlo (QMC) methods for generating training data for neural networks, with a particular emphasis on convergence behavior in both deterministic and stochastic settings. In contrast to classical Monte Carlo (MC) approaches, which rely on pseudorandom sampling, QMC methods use low-discrepancy sequences to achieve more uniform coverage of the input space, which can lead to improved approximation performance, especially in high-dimensional problems.

In the first part of the thesis, QMC methods are applied to sample training inputs from a bounded domain. The corresponding function values are computed and the resulting dataset is used to train neural networks. The convergence of the network output with respect to the number of training points is analyzed and compared against networks trained on MC-sampled data. The goal is to quantify the advantage of QMC-based training in terms of learning efficiency and approximation accuracy.

The second part extends this investigation to stochastic functions. Here, the randomness in the system—such as in photon transport models—is explicitly modeled and both QMC and MC sampling are used to sample from the stochastic variables. The focus is on physically motivated functions that simulate scattered photon radiation, a key challenge in medical imaging applications such as x-ray or CT simulations. Neural networks are trained on data generated from these stochastic models and their convergence behavior is evaluated under QMC and MC sampling regimes.

Through these experiments, the thesis demonstrates how QMC-based sampling can enhance neural network training, particularly when approximating complex physical processes. The results provide both theoretical insights and empirical evidence that QMC methods offer significant advantages over standard MC approaches in terms of convergence speed and generalization quality in high-dimensional learning problems.

Contents

RITA Rational Inverse Transform with Aliasing

CDF Cumulative Distribution Function

QMC Quasi-Monte Carlo

MC Monte Carlo

HU Hounsfield Unit

CT Computed Tomography

CBCT Cone-Beam Computed Tomography

FFD Forced Fixed Detection

Part I

X-ray Simulation using QMC Methods

Chapter 1

Motivation and Problem Statement

1.1 Computed Tomography Imaging

Computed Tomography (CT) imaging is a powerful medical imaging technique that provides detailed cross-sectional images of a body by combining multiple X-ray projections taken from different angles. This technique is widely used for diagnostic purposes, allowing clinicians to visualize internal structures with high spatial resolution.

The process involves rotating an X-ray source around the patient while simultaneously capturing X-ray projections on a detector array. Each projection represents the cumulative attenuation of X-rays as they pass through various tissues, which is influenced by the density and composition of the materials they encounter.

The collected data is then reconstructed into a three-dimensional volume using sophisticated algorithms, such as filtered backprojection or iterative reconstruction methods. These algorithms convert the raw projection data into cross-sectional images, which can be further processed to enhance contrast, reduce noise and improve overall image quality.

To achieve accurate reconstructions, it is crucial to have precise and high-quality X-ray images. However, one of the most significant challenges in CT imaging is the presence of scattered radiation, which can lead to artifacts and distortions in the reconstructed images.

Throughout this thesis, the term *scatter reduction in CT* refers to the mitigation of scatter-induced artifacts in the 2D X-ray projection images acquired by the detector array during a CT scan. These projections serve as the input for reconstructing the final 3D volume. Given the complexity of the full reconstruction process, the focus of this work will be limited to analyzing and correcting scatter effects at the level of the 2D projection data.

1.2 X-ray Imaging and the Challenge of Scatter

X-ray CT relies on measuring the attenuation of X-ray photons as they traverse straight-line paths through a phantom. These trajectories typically extend from an X-ray source S to a detector element $\mathcal{D}_j$, forming the path:

$$l = \overrightarrow{S\mathcal{D}_j}$$

Under idealized conditions, the attenuation process is goverend by the Lambert-Beer law, which describes an exponential decay in intensity as the beam interacts with the material. As stated in [3, Chap. 7] ,this relationship is given by

$$I = \int_0^{E_{\max}} I_0(E) \cdot \exp\left(- \int_S^{\mathcal{D}_j} \mu(x, E) dx \right) dE \tag{1.1}$$

where $I_0(E)$ denotes the incident intensity of X-rays with energy E at source S. Further $\mu(x, E)$ represents the linear attenuation coefficient of a photon at any spatial location x along the path l with energy E. The resulting intensity I is measured at the detector element $\mathcal{D}_j$.

Although exponential attenuation along straight-line paths is the idealized model for X-ray imaging, this assumption is systematically violated in practice. As X-rays traverse the scanned object, many photons undergo scattering interactions - such as Compton or Rayleigh scattering - which alter their trajectories. Despite deviating from the primary path, these scattered photons may still reach the detector, adding unintended signal components. As a result, the measured intensities no longer represent pure line integrals of the attenuation map. This discrepancy introduces nonlinear errors and visible artifacts in the reconstructed image, ultimately degrading both visual quality and quantitative accuracy.

Scattered radiation is a major source of image artifacts - such as cupping and streaks - and reduces both spatial and contrast resolution. These effects not only degrade visual image quality but also compromise the accuracy of quantitative measurements, such as Hounsfield units μ_* [3, Chap. 8], which are used for clinical interpretation such as tissue characterization. Hounsfield units are representing a normalized attenuation to the attenuation of water.

$$\mu_* = \left(\frac{\mu}{\mu_{\text{water}}} - 1 \right) \cdot 1000$$

The impact of scatter becomes especially pronounced in modern CT systems using high-energy X-rays or large-area flat-panel detectors, where scatter may dominate the measured signal. As such, accurate modeling and correction of scatter is essential for achieving high-fidelity CT images, particularly in clinical applications where precision and reliability are paramount [3].

1.3 Scatter Correction Methods for X-ray Imaging

1.3.1 Overview of Scatter Correction Techniques

In CT, scattered photons are a major source of image artifacts and quantitative inaccuracies. To mitigate these effects, a range of computational scatter correction methods has been developed. These approaches can be broadly classified into the following categories:

- **Empirical and Analytical Methods:**
  These include techniques such as primary modulation, convolution-based correction and energy windowing. Scatter is typically estimated using simplified models or empirical kernels, often assuming a smooth background distribution and then subtracted from the measured signal. While computationally

efficient, these methods rely on assumptions regarding the spatial and energy distribution of scattered photons. As a result, they may fail to accurately model complex scatter phenomena in heterogeneous anatomical structures.

- **Physics-Based Models:**
  These methods aim to provide a more accurate representation of the underlying photon transport physics, including scattering phenomena. Among them, Monte Carlo (MC) simulation is regarded as the most rigorous and comprehensive technique due to its ability to statistically model complex photon interactions without relying on simplifying assumptions. In such models, primary and scattered photon contributions are simulated separately, allowing the estimated scatter signal to be subtracted from measured data in order to restore image fidelity.

In 2011 Sisniega et al. [9] already demonstrated promising results using the computationally expensive Monte Carlo (MC) simulation of the CT of a cylinder phantom with two bone inserts. The results show significant improvements in the image quality as demonstrated by Sisniega et al. in Figure 1.1.

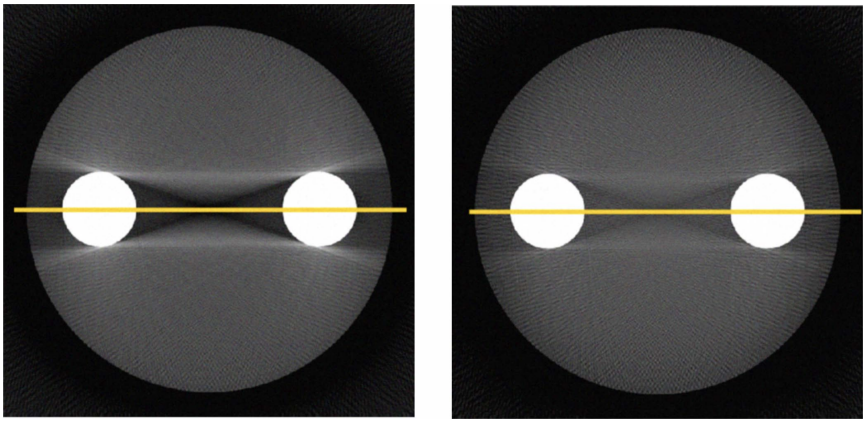


FIGURE 1.1: Cone-Beam Computed Tomography (CBCT) of a 70 mm soft-tissue cylinder with two bone inserts. Left: Image after scatter correction using the gMCFFD method (5*e*6 photons), showing improved uniformity and quantitative accuracy. Right: Image without correction, exhibiting typical cupping and streak artifacts. (from [9])

1.3.2 Monte Carlo Simulation

To achieve scatter correction, by using MC simulation, first a three-dimensional phantom is estimated by analysing the X-ray projection. Then a MC simulation is performed to estimate the scatter signal $I_{\text{scattered}}$ for each detector pixel. Then the estimated scatter signal is subtracted from the measured intensity I_{measured}:

$$I_{\text{corrected}} = I_{\text{measured}} - I_{\text{scattered}} \tag{1.2}$$

The functionality of MC based scatter correction is based on the following key steps:

The MC simulation is a powerful computational technique that follows phyics-based principles to model the transport of photons through matter. It is widely recognized as the gold standard for scatter correction in CT and related imaging modalities. This status is attributed to several key factors:

- **Physical Accuracy:**
  MC methods simulate the stochastic nature of photon interactions (including Compton and Rayleigh scattering, photoelectric absorption and multiple scattering events) based on fundamental physical cross-sections and material properties.
- **Comprehensive Modeling:**
  Unlike analytical or empirical methods, MC simulation can account for complex geometries, heterogeneous materials and realistic X-ray spectra, providing highly accurate estimates of the scatter signal.
- **Validation Benchmark:**
  Due to their accuracy, MC-based scatter estimates are routinely used as reference standards for validating and benchmarking faster, approximate correction methods.

1.4 High-Level Overview of the Monte Carlo Simulation

Monte Carlo simulation of photon transport for scatter correction involves the following key steps [9]:

1. **Photon Emission:**
   Photons are emitted from a virtual X-ray source, with energies sampled from the source spectrum.
2. **Photon Tracking:**
   Each photon is tracked as it propagates through the object. At each step, the probability of interaction (scattering or absorption) is determined by the local material properties and cross-sections.
3. **Interaction Sampling:**
   When an interaction occurs, the type (e.g., Compton, Rayleigh or Absorption) and the resulting change in photon direction and energy are sampled from the relevant probability distributions.
4. **Detection:**
   Photons that reach the detector (either unscattered or after one or more scatter events) are recorded. The simulation keeps track of both primary and scattered photons, allowing for the estimation of the scatter contribution to each detector element.
5. **Statistical Averaging:**
   By simulating a large number of photons the method builds up a statistical robust estimate of the scatter distribution. The accuracy of the result increases with the number of simulated photons and eventually converges to a stable intensity distribution.

The schematic flow of the Monte Carlo simulation process is illustrated in Figure 1.2.

Monte Carlo Photon Transport Algorithm

- **Initialization:** Define geometry, materials and source spectrum.
- **Loop over photons:**
  - Sample step size to next interaction.
  - Move photon; check for boundary crossing.
  - Sample interaction type; update direction/energy.
  - If photon reaches detector, record event.
  - If photon is absorbed or leaves the system, terminate.
- **Aggregate results:** Compute scatter and primary distributions.

FIGURE 1.2: Pseudocode for photon transport simulation in Monte Carlo scatter modeling.

1.5 Monte Carlo Methods: Benefits & Drawbacks

Monte Carlo (MC) simulations are widely regarded as the gold standard for scatter correction in computed tomography due to their ability to model photon-matter interactions from first principles. These simulations faithfully reproduce all relevant physical scattering phenomena - including Compton and Rayleigh scattering - and can accommodate arbitrarily complex object geometries and heterogeneous material compositions. Owing to this high level of physical realism, MC-based methods yield highly accurate scatter estimates and are therefore commonly used as a reference benchmark for evaluating and validating alternative correction approaches.

However, the high accuracy of Monte Carlo simulations comes at the cost of substantial computational demand. Achieving low-noise scatter estimates requires simulating a large number of photon histories to ensure statistical convergence. As a result, the runtime scales with the desired level of accuracy and can span from several minutes to multiple hours or even days, depending on the complexity of the problem and the available computational resources. This computational burden poses a major practical limitation, particularly in scenarios involving large datasets or iterative reconstruction workflows.

In response to the computational burden of slow converging traditional Monte Carlo methods, recent research has focused on approaches to accelerate this type of precisely correct photon transport simulations. Among these, the application Quasi-Monte Carlo (QMC) methods has gained increasing attention. By replacing random sampling with deterministic low-discrepancy sequences, QMC techniques achieve significantly faster convergence while maintaining comparable accuracy. Contemporary QMC-based scatter correction algorithms have demonstrated runtime reductions of one to two orders of magnitude, thereby enabling high-fidelity simulations even in time-sensitive or resource-constrained settings [2].

Chapter 2

Physical laws of Photon Simulation

This chapter introduces the physical and mathematical principles underlying the simulation of X-ray imaging. Central to this is the modeling of individual photon interactions with matter, including attenuation and scattering processes. These interactions are inherently stochastic due to the quantum nature of radiation-matter interactions. The stochastic nature of photons and their interactions with matter is modelled using monte Carlo methods of uniformly distributed values between 0 and 1, which are then transformed to follow the physical probability distributions relevant to the specific interactions being simulated. This approach allows for a realistic representation of photon transport and interaction within the imaging system.

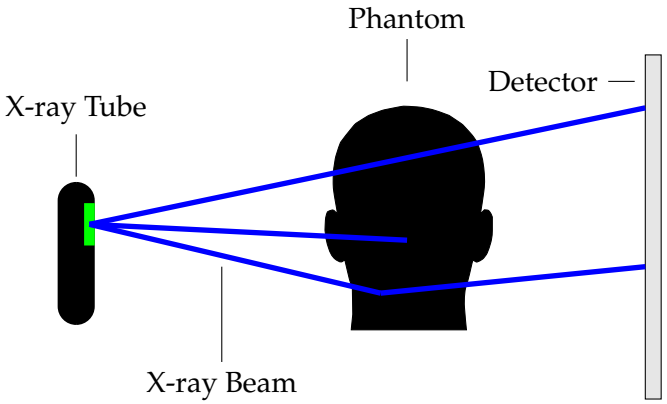


FIGURE 2.1: Schematic illustration of photon transport in X-ray imaging.

In a typical simulation workflow, individual photons are emitted from the X-ray tube and propagate through air before reaching the phantom. Upon entering the phantom, they may interact with the material through scattering or absorption, depending on the local composition of the tissue and photon energy. Only a fraction of the photons will exit the phantom, continue their path through air, and ultimately reach the detector, contributing to the final image formation.

The simulation framework described here forms the basis for all subsequent analyses presented in this thesis.

2.1 X-Ray Tube

X-ray beams are generated within evacuated glass tubes containing several critical components that convert electrical energy into X-ray photons and heat. The X-ray tube acts as an energy converter, where the vast majority of the input energy is transformed into heat and only a small fraction becomes useful radiation. The following Figure 2.2 illustrates the basic construction of an X-ray tube as in [7]

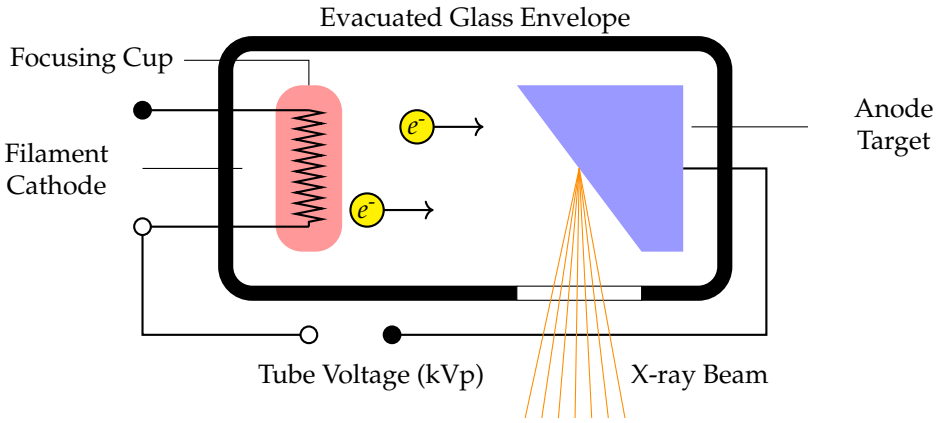


FIGURE 2.2: Schematic of an X-ray tube.

2.1.1 Photon Generation

To capture this randomness, Monte Carlo methods are employed. In such simulations, random numbers—typically sampled uniformly from the interval $[0, 1]$ are transformed into physically meaningful quantities according to the relevant probability distributions. This probabilistic modeling enables realistic simulation of photon transport and forms the basis for the analyses presented in this thesis.

The key components are:

- **Filament (Cathode):** A tungsten wire filament serves as the electron source through thermionic emission. When a current of approximately 3-6 A passes through it, the filament reaches incandescence, releasing electrons from its surface. These free electrons form a cloud near the cathode until they are accelerated toward the anode by the applied high voltage.
- **Focusing Cup:** The filament is embedded in a negatively charged, nickel-made focusing cup. Its function is to electrostatically shape and direct the electron stream toward the anode's focal spot, thereby influencing the resolution and size of the resulting X-ray beam.
- **Anode Target:** The anode consists of a tungsten target, often embedded in a copper support. Tungsten is used for its high atomic number and melting point, enhancing X-ray production and durability. The copper base improves heat dissipation. Typically, less than 1% of the electron energy is converted into X-rays, with the remainder generating heat that must be effectively managed.
- **Evacuated Glass Envelope:** All components are sealed within a borosilicate glass or metal-ceramic housing evacuated to a low pressure (typically 10^{-5} to 10^{-7} hPa). The vacuum allows unimpeded electron flow and prevents arcing.

The housing is usually immersed in insulating oil to provide thermal and electrical isolation.

When high-energy electrons strike the tungsten target, X-ray photons are produced through two primary mechanisms:

- **Bremsstrahlung (Braking Radiation):** This accounts for approximately 80% of X-ray production. When electrons pass close to tungsten nuclei, they are decelerated by the electrostatic attraction, causing them to lose kinetic energy that is emitted as X-ray photons. This process produces a continuous spectrum of X-ray energies from near zero up to the maximum electron energy.
- **Characteristic Radiation:** This occurs when high-energy electrons knock inner shell electrons from tungsten atoms. When outer shell electrons drop down to fill these vacancies, they emit X-rays with discrete, characteristic energies specific to tungsten. Therefore peaks are occuring at the difference of the binding energies of the electron shells. For reference, the atomic model of tungsten is given in Figure 2.7 and the approximate binding energies of the electron shells in Table 2.1.

Shell / Subshell	Binding Energy (keV)
K	69.5
L_1	12.1
L_2	11.5
L_3	10.2
M_1	2.82
M_2	2.30
M_3	2.15
N_1	0.43
N_2	0.32
N_3	0.22

TABLE 2.1: Approximate Electron Binding Energies of Tungsten (W, Z = 74)

In Table 2.1 only binding energies of the K, L and M shells are given, since these are the most relevant for X-ray production. The binding energies of the N shell and higher shells are negligible in this context due to their low binding energies.

Although it is not necessary to understand every technical detail of the X-ray apparatus for the purposes of simulation, I found it important to briefly present the fundamental working principles of the X-ray tube. This background allows one to appreciate how key simulation parameters for photon generation - specifically the tube voltage and the cathode material - influence the resulting X-ray spectrum and photon behavior.

Figure 2.4 below shows an resulting energy spectrum for an X-ray tube with a tungsten cathode operated at 100 kVp. It illustrates the resulting distribution of photon energies, which is shaped by both the material and the applied voltage. The intensity of the different photon energies is measured in *spectral fluence* in $\mathrm{cm}^{-2}\,\mathrm{keV}^{-1}$,

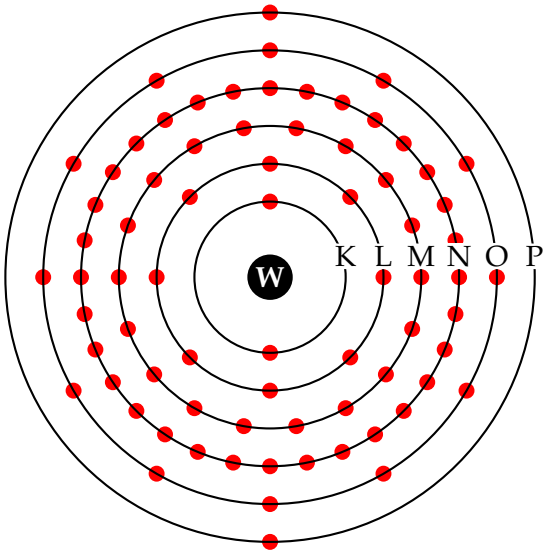


FIGURE 2.3: Bohr model of the tungsten atom (W, Z = 74) with electron shells and subshells.

which describes the number of X-ray photons per unit area per unit energy interval. It certainly shows the two components of the X-ray spectrum: the continuous bremsstrahlung spectrum and the characteristic radiation peaks:

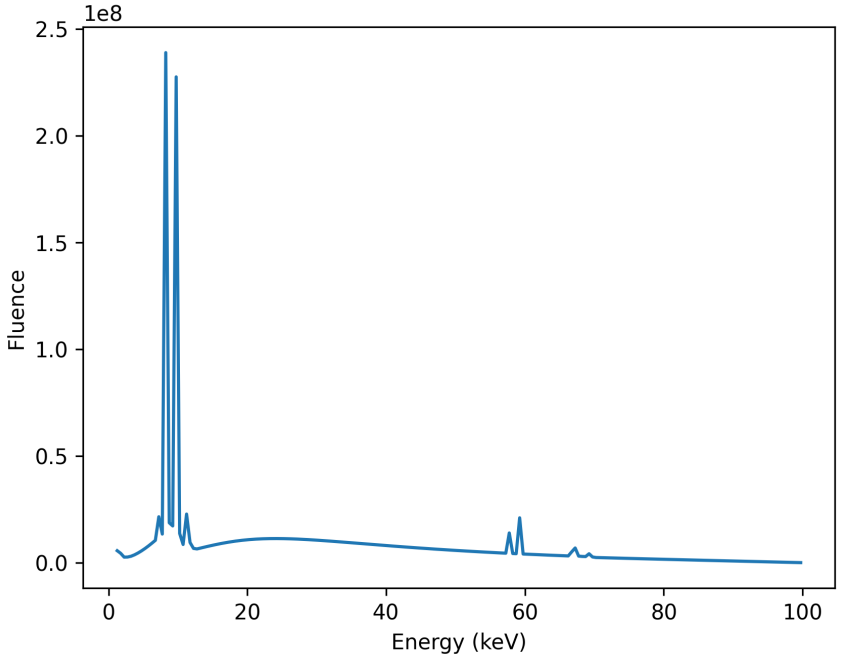


FIGURE 2.4: X-ray spectrum for a tungsten cathode at 100 kVp showing the continuous bremsstrahlung spectrum and characteristic peaks. Build with *SpekPy* [6]

2.1.2 Filter

Low energy X-ray photons contribute little to image formation but significantly increase patient dose. Therefore, X-ray tubes are often equipped with filters to selectively attenuate these low-energy photons while allowing higher-energy photons to pass through. This process, known as beam hardening, improves image quality by reducing scatter and enhancing contrast. Common filter materials include aluminum, copper and molybdenum, which are chosen based on their atomic number and thickness to effectively absorb low-energy photons while minimizing the impact on higher-energy photons [7].

As can be seen in Figure 2.5, the filter is placed at the X-ray tube margin such that the photons hit the filter after the anode target and before leaving the X-ray tube. Followingly X-rays pass through the filter before reaching the patient.

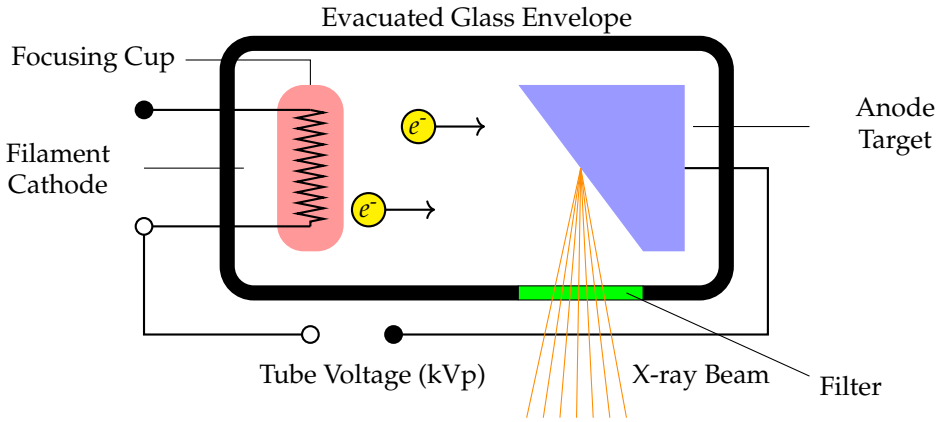


FIGURE 2.5: Schematic of an X-ray tube with Filter.

Throughout this thesis, a filter consisting of 0.4 mm Tin (Sn) and 0.1 mm Copper (Cu) is used for the simulations. This filter is chosen to represent a realistic X-ray tube filter that is commonly used in clinical practice [10]. The resulting spectrum is shown in Figure 2.6.

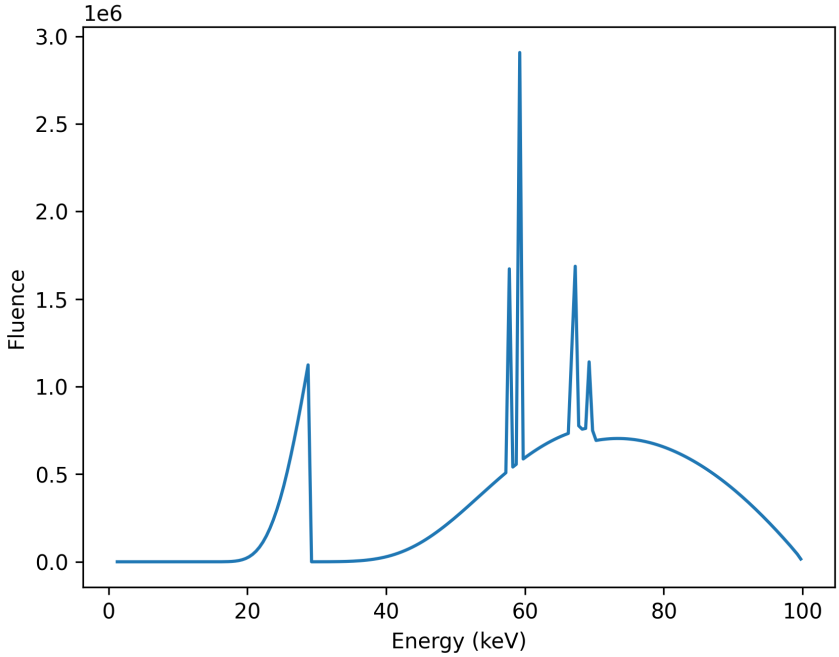


FIGURE 2.6: X-ray spectrum for a tungsten cathode at 100 kVp with a 0.4 mm showing the continuous bremsstrahlung spectrum and characteristic peaks. Build with *SpekPy* [6]

2.2 Photon Generation

In the context of Monte Carlo simulations, photons are generated from the X-ray tube source, which is typically modeled as a point source emitting photons within a conical beam. The emission cone is defined by a half-angle Θ, which determines the angular distribution of emitted photons. – The emitted photons are characterized by two key properties: their direction and energy.

Photon Energy

The photon energy is sampled from the X-ray tube spectrum as described in Section 2.1.2. In the simulations referenced in this thesis the spectra are generated utilizing *SpekPy* [6, 8]. The energies of the resulting photons are sampled via inverse transform sampling from the normalized spectrum.

Inverse transform sampling [4] is a method to generate random samples from a target distribution using uniformly distributed random variables, such as those generated by a Quasi-Monte Carlo sequence.

Definition 2.2.1 (Inverse Transform Sampling)**.**
Let $F_X : \mathbb{R} \to [0,1]$ be the cumulative distribution function (CDF) of a continuous, strictly increasing random variable X. The method of *inverse transform sampling* generates a realization of X by the following procedure:

1. Generate a sample U from the uniform distribution on the unit interval, i.e., $U \sim \mathcal{U}(0,1)$.

2. Compute the value $X := F_X^{-1}(U)$, where F_X^{-1} denotes the inverse of the CDF F_X.

Theorem 2.2.2.
Let F_X be a continuous and strictly increasing cumulative distribution function and let $U \sim \mathcal{U}(0,1)$. Then the random variable

$$X := F_X^{-1}(U)$$

has cumulative distribution function F_X, i.e.,

$$\mathbb{P}(X \leq x) = F_X(x), \quad \textit{for all } x \in \mathbb{R}.$$

Proof.
Since F_X is continuous and strictly increasing, its inverse F_X^{-1} exists. For any $x \in \mathbb{R}$, we compute:

$$\mathbb{P}(X \leq x) = \mathbb{P}(F_X^{-1}(U) \leq x) = \mathbb{P}(U \leq F_X(x)) \overset{\text{since } F_X \text{ is strictly decreasing}}{=} F_X(x),$$

because $U \sim \mathcal{U}(0,1)$ and thus $\mathbb{P}(U \leq u) = u$ for all $u \in [0,1]$. This shows that X has CDF F_X. □

Photon Direction

The direction of each emitted photon is sampled uniformly within a conical emission cone defined by the half-angle Θ. For the simulation a spherical alignment of the X-ray tube is assumed, such that the beam is oriented along the vector $\vec{v} = (v_1, v_2, v_3)$ in the Cartesian coordinate system.

Followingly, the direction of the emitted photon is sampled based on two random values $u_1, u_2 \in [0,1)$ as follows:

1. Sample a random angle θ uniformly from the interval $[0, \Theta]$ with u_1:

$$\theta = u_1 \cdot \Theta$$

2. Sample a random azimuthal angle ϕ uniformly from the interval $[0, 2\pi)$ with u_2:

$$\phi = u_2 \cdot 2\pi$$

3. Compute the direction vector $\vec{d}$ of the photon as:

$$\vec{d} = (\sin(\theta)\cos(\phi), \sin(\theta)\sin(\phi), \cos(\theta))$$

2.3 Scattering and Attenuation

This section provides the basic concepts of photon interaction with matter. The interaction relevant for medical imaging results in a reduction of radiation intensity, which corresponds to a decreased number of photons reaching the detector. Hereby X-ray photons may be fully absorbed by *photoelectric absorption* or undergo either

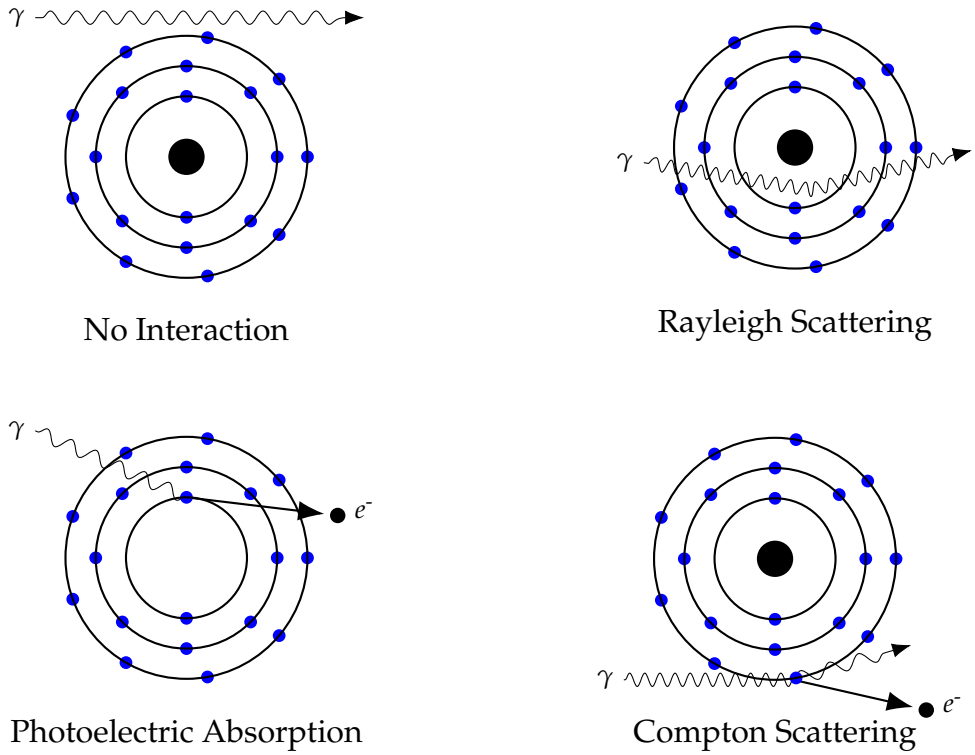


FIGURE 2.7: Principles from photon interaction with matter similar to [3, Chap. 7]

elastic scattering (Rayleigh) or *inelastic scattering* (Compton) as they interact with matter.

The attenuation of X-ray photons arises from physical processes that alter their number, direction, or energy as they interact with matter. These interactions occur at the level of individual photons and are highly dependent on the photon energy. This section presents an overview of the primary interaction mechanisms relevant to attenuation like in [3].

For correctness, in the simmulations it is further assumed that the X-ray photons are propagating through vacuum before entering and after exiting the phantom. Usually the tissues are surrounded by air, which has a negligible effect on the photon transport.

2.3.1 Free Path Length

All mentioned interaction processes - the photoelectric effect, Compton scattering and Rayleigh scattering - are probabilistic in nature. The according attenuation coefficients μ characterizes the extend of the beam being reduced by its according effect, when passing through the material, in $\mathrm{cm}^2\,\mathrm{g}^{-1}$. For the simulation of photons, the attenuation coefficients are dependend on the according energy E of the photon and the material at spherical location x of the photon. The coefficients are summarized in Table 2.2.

Interaction Type	Attenuation Coefficient
Photoelectric Effect	$\mu_{\text{PE}}(x, E)$
Rayleigh Scattering	$\mu_{\text{RS}}(x, E)$
Compton Scattering	$\mu_{\text{CS}}(x, E)$

TABLE 2.2: Attenuation coefficients for different photon interaction mechanisms.

The linear attenuation coefficient $\mu(x, E)$ is the sum of the individual attenuation coefficients of the interaction coefficients:

$$\mu(x, E) = \mu_{\text{PE}}(x, E) + \mu_{\text{RS}}(x, E) + \mu_{\text{CS}}(x, E)$$

When an X-ray photon eneters the phantom two main effects are considered:

- **Attenuation:** The photon may be absorbed or Scattered.
- **No Interaction:** The photon may pass through the phantom without any interaction.

The *free path length* t of a photon describes the distance a photon takes to traverse through the phantom before it interacts with matter. Supposing the phantoms location is x and the photon is traveling in the unit direction $\vec{v}$, as in [2] the free path length t follows the distribution given by:

$$t \sim \mu(x, E) \exp\left[-\int_0^t \mu(x + s \cdot \vec{v}) \right]$$

Check: nach dem paper müsste μ abhängig vom Endpunkt sein.

The free path length t is constraint by the maximum distance c ($0 \leq t \leq c$) to the next exit point of the phantom along the ray with direction $\vec{v}$. After leaving the phantom, we assume the photon is reaching the detector or leaving the simulation domain.

The probability of the photon to leave the phantom unaltered is described by the *escape probability* $\mathcal{P}$ of a photon at a given position A with unit direction $\vec{v}$ and energy E as in [2]:

$$\mathcal{P}(x, \vec{v}, E) = \exp\left[-\int_0^c \mu(x + t\vec{v}, E) dt \right],$$

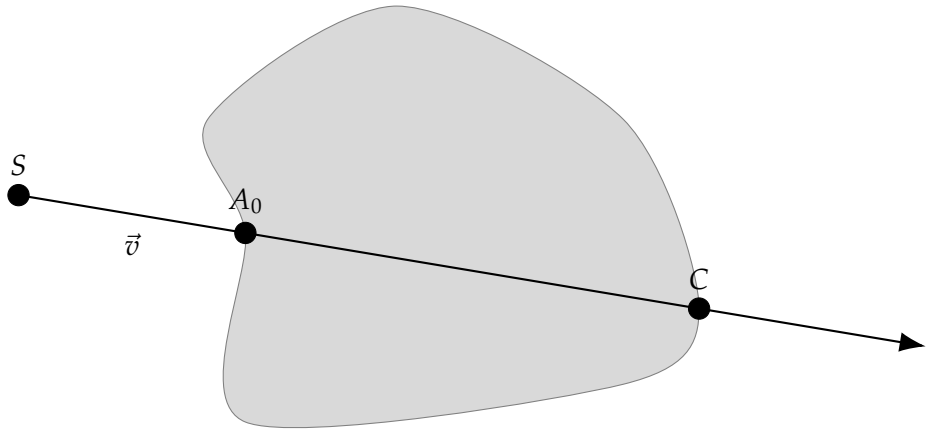


FIGURE 2.8: Illustration of the entry and exit point of the ray of a photon without interaction.

2.3.2 Compton Scattering

Compton scattering is the most dominant interaction mechanism for X-ray photons in tissue [3, Chap. 7]. It occurs when a X-ray photon with considerably higher energy than the binding energy of an outer shell electron collides with this electron. This interaction results in a transfer of energy and momentum. The electron is ejected from the atom, while the incoming photon is scattered at an angle and its energy is reduced.

A portion of the incident photon energy is transferred to the electron, which is referred to as the "recoil electron" or Compton electron. The interaction produces a positive ion, the "recoil electron" and a scattered photon. If the deflection angle of the scattered photon is small, most of the energy is retained by the scattered photon. The deflection angle can vary from 0 to 180 degrees, depending on the energy transfer during the interaction.

2.3.3 Rayleigh Scattering

Rayleigh scattering is a type of elastic scattering that occurs at low X-ray energies [3, Chap. 7]. The incident photon interacts with several electrons that are usually bound in the outer shells of the atom. In this process, the low energy photon is not ejected, but rather the electrons and in turn thw whole atom is set to vibration with respect to the incident photon's wavelength. The vibrating photon transfers its excess energy to an electromagnetic photon with the same wavelength but probably a different direction than the incident photon. The majority of these scattered photons are emitted in a forward direction. This interaction does not result in the ejection of electrons from the atom and no ionization occurs as no energy is converted into kinetic energy.

Although Rayleigh scattering is taking place in the X-ray tube, it can be excluded from the linear attenuation coefficient, as it does not contribute to the attenuation of the X-ray beam in the same way as Compton scattering or photoelectric absorption. As described in [7], the exclusion of Rayleigh scattering from the linear attenuation coefficient is based on the assumption that the characteristic radiation emitted by

the target is isotropic, meaning it is emitted uniformly in all directions. As the X-ray is not attenuated by Rayleigh scattering, it is a common assumption to exclude Rayleigh scattering from the attenuation coefficient.

Chapter 3

The Simulation Algorithm

Chapter 4

The Simulation Algorithm

This chapter presents the algorithm used for X-ray image simulation. It incorporates elements of *Forced Detection* as introduced in [1], in order to accelerate convergence.

Unlike standard Monte Carlo simulations, where a random variable determines whether a photon escapes the phantom or undergoes another scattering event, the forced detection approach employed, calculates the probability of escaping the phantom at each interaction point and reflects this distribution in accordingly updated photon intensities for the case of escaping the phantom and the case of scattering. Instead of probabilistically terminating the photon trajectory, the algorithm tracks it through a predefined number of scattering events N. At each interaction, the remaining intensity is updated to reflect both the probability of Compton scattering and the conditional escape probability. This ensures that both potential outcomes—escape or continued scattering—are implicitly accounted for in the evolving photon intensity. As a result, the simulation maintains physical accuracy while achieving a significant reduction in variance.

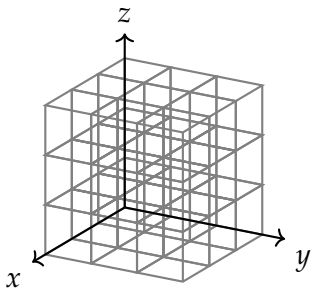
When a photon eventually escapes the phantom, it contributes to the corresponding detector pixel, weighted by its current intensity, photon energy, and escape probability. Conversely, the distance to the next interaction is sampled based on the probability of not escaping the phantom. Further implementation details are discussed in Section 4.1.

This variance reduction strategy enables the simulation to converge more rapidly toward high-resolution images with suppressed noise and well-preserved structural detail.

4.1 Algorithm overview

The algorithm begins by initializing a set of photons, each assigned an initial energy E_0, sampled from the X-ray tube spectrum and an initial direction $\vec{\omega}_0$ sampled uniformly within a cone centered around the principal beam axis. A total of three random variables are drawn for each photon to determine its energy and direction.

Given a fixed source position A representing the location of the X-ray tube, each photon's initial origin is placed within a voxel grid composed of cubic voxels that define the geometry of the scene. Each voxel encodes an integer value identifying a specific material or tissue type. Furthermore, each photon is initialized with an intensity of $W_0 = 1$, which is iteratively updated throughout the simulation to account for attenuation and scattering processes.



Then an algorithm is applied to traverse the photon through the voxel grid until it reaches the first material which is not air. This is done with a ray traversal algorithm. Once the phantom at a point A_0 is reached, the photon transport is simulated by the physical laws described in Chapter 1.

Hereby, as in all other scatter points within the phantom, the free *free path length* t_i is sampled from the exponential distribution based on *Beer-Lambert's law* (Eq. 1.1) and the *total attenuation coefficient* $\mu(A_i, E_i)$ accordingly. A_i hereby denotes the current position and E_i the current energy. First, the *escape probability* depending on the current position A_i and the unit direction $\vec{\omega}_i$ and the energy E_i is computed. The escape probability in Equation 4.1 is the probability of the photon escaping the phantom at the current position A_i without being further scattered.

$$p(A_i, \omega_i, E_i) = \exp\left(- \int_{\overrightarrow{A_i C_i}} \mu(x, E_i) dx \right) \tag{4.1}$$

Hereby C_i denotes the exit point of the phantom in the direction of the photon $\vec{\omega}_i$. The escape probability is used to determine the intensity of the photon without the $(i+1)$-th scatter event.

Using one random variable u_{3i+1}, the free path length is sampled. Hereby both cases are being considered:

1. **Photon escapes the phantom without further scattering:**
   This happens with the portion matching the escape probability $p(A_i, \omega_i, E_i)$. Accordingly
   $$W_{i+1}^{\text{escape}} = W_i \cdot p(A_i, \omega_i, E_i) \tag{4.2}$$
   The photon contributes to the detector signal in direction $\vec{\omega}_i$ with a final intensity of $E_i \cdot W_{i+1}^{\text{escape}}$. In case $i = 0$, this intensity is accounted as priamry intensity.

2. **Photon does not escape the phantom and scatters:**
   This happens with the portion matching the inverse of the escape probability $1 - p(A_i, \omega_i, E_i)$. Accordingly, the free path length is sampled from the exponential distribution by solving Equation 4.3 for t_i:
   $$\exp\left(- \int_0^{t_i} \mu(A_i + s \cdot \vec{\omega}_i, E_i) ds \right) = u_{3i+4} \tag{4.3}$$

Accordingly, the next scatter point is being computed as:

$$A_{i+1} = A_i + t_i \cdot \vec{\omega}_i \tag{4.4}$$

The photon intensity is then updated with:

$$W_{i+1} = W_i \cdot (1 - p(A_i, \omega_i, E_i)) \cdot \frac{\mu_{\text{CS}}(A_{i+1}, E_i)}{\mu(A_{i+1}, E_i)} \tag{4.5}$$

In the next step, the photon energy and opening angle after the Compton scatter event is sampled with a second random variable u_{3i+2}. A thrid variable is used to apply a azimuthal rotation around the direction of the photon $\vec{\omega}_i$ to sample the new direction $\vec{\omega}_{i+1}$ of the photon after the scatter event.

This process is repeated according to the maximum scatter order N.

4.2 Sub-Algorithms

To model the relevant physical processes in a structured manner, the main simulation is decomposed into several sub-algorithms. This section describes the purpose and implementation of each sub-algorithm in detail.

4.2.1 Photon Generation

For the photon generation step, two fundamental properties are sampled for each photon:

- *Photon energy*, sampled using a single random variable from the X-ray spectrum.
- *Photon direction*, sampled using two random variables uniformly within a cone defined by the beam axis $\vec{d}$ and opening angle α.

Photon Energy Sampling

The photon energy is sampled from the X-ray tube spectrum, which is represented as a discrete set of energy values with corresponding fluence values. The spectrum was generated using *Spekpy* [6, 8], based on an X-ray tube configured with the parameters listed in Table 4.1.

Parameter	Value
Tube voltage	120 kV
Anode material	Tungsten
Filtration	0.4 mm Tin (Sn), 0.1 mm Copper (Cu)
Target angle	12.5°

TABLE 4.1: Parameters used to generate the X-ray spectrum with *Spekpy*.

Photon energies are sampled using inverse transform sampling. The algorithm normalizes the fluence values to obtain a probability density function (PDF), computes

the corresponding cumulative distribution function (CDF) and uses a uniformly distributed random variable to select an energy according to the CDF.

Algorithm 1 Photon Energy Sampling from Spectrum

Require: Array of energies $E = [E_1, E_2, \ldots, E_n]$
Require: Corresponding fluence values $\Phi = [\phi_1, \phi_2, \ldots, \phi_n]$
Require: Number of samples N
Require: Random variable $u \sim \mathcal{U}(0,1)$
Ensure: Sampled photon energies $S = [s_1, s_2, \ldots, s_N]$
// Normalize fluence values:
1: $$T \leftarrow \sum_{i=1}^{n} \phi_i$$
2: $$\text{PDF}[i] \leftarrow \frac{\phi_i}{T}$$
// Compute cumulative distribution function:
3: $$\text{CDF}[1] \leftarrow \text{PDF}[1]$$
4: **for** $i = 2$ to n **do**
5: $\quad \text{CDF}[i] \leftarrow \text{CDF}[i-1] + \text{PDF}[i]$
6: **end for**
// Note: $PDF[i] > 0$ in the spectrum, therefore CDF is strictly increasing.
7: Create interpolating function $\text{InverseCDF}(u)$ from $(\text{CDF}[i], E[i])$
8: **return** $\text{InverseCDF}(u)$

Photon Direction Sampling

The direction of each photon is sampled uniformly within a cone defined by the beam axis $\vec{d}$ and opening angle α. This requires two independent random variables $u_1, u_2 \sim \mathcal{U}(0,1)$.

Algorithm 2, adapted from [11], generates a random unit vector $\vec{v}$ uniformly distributed within a cone of half-angle α around the direction $\vec{d}$:

1. **Sampling the polar angle** θ:
   Draw $u_1 \sim \mathcal{U}(0,1)$ and compute
   $$\theta = \arccos\left(1 - u_1(1 - \cos\alpha)\right).$$
   This corresponds to inverse transform sampling such that $\cos\theta$ is uniformly distributed on $[\cos\alpha, 1]$, resulting in uniform sampling over the spherical cap.

2. **Sampling the azimuthal angle** ϕ:
   Draw $u_2 \sim \mathcal{U}(0,1)$ and compute
   $$\phi = 2\pi u_2,$$
   ensuring a uniform distribution around the cone axis.

3. **Constructing the local direction vector**:
   In a local spherical coordinate system with the cone axis aligned along the z-axis, the sampled unit vector is

$$\vec{v}_{\text{local}} = \begin{pmatrix} \sin\theta\cos\phi \\ \sin\theta\sin\phi \\ \cos\theta \end{pmatrix}.$$

4. **Rotation into global coordinates**:
   To align the cone axis from the local z-axis to an arbitrary unit vector $\vec{d}$, an orthonormal basis $(\vec{u}, \vec{v}, \vec{d})$ is constructed via the Gram–Schmidt process:
   - Choose a helper vector $\vec{a} = (0,0,1)$ if $|d_3| < 0.999$, otherwise $\vec{a} = (1,0,0)$.
   - Compute $\vec{u} = \frac{\vec{a}\times\vec{d}}{\|\vec{a}\times\vec{d}\|}$.
   - Set $\vec{v} = \vec{d} \times \vec{u}$ to complete a right-handed orthonormal basis.

   Finally, rotate $\vec{v}_{\text{local}}$ into global coordinates via the transformation

$$\vec{v}_{\text{global}} = (\vec{v}_{\text{local}})_x\vec{u} + (\vec{v}_{\text{local}})_y\vec{v} + (\vec{v}_{\text{local}})_z\vec{d}.$$

Algorithm 2 Uniform Direction Sampling Within a Cone

Require: Cone angle α
Require: Unit beam direction vector $\vec{d} = (d_1, d_2, d_3)$
Require: Random variables $u_1, u_2 \sim \mathcal{U}(0,1)$
Ensure: Sampled direction vector $\vec{v}$ uniformly within cone around $\vec{d}$

// Calculate angles according to samples
$\theta \leftarrow \arccos(1 - u_1(1 - \cos\alpha))$ // Polar angle
$\phi \leftarrow 2\pi u_2$ // Azimuthal angle
// Calculate local direction vector in spherical coordinates

$$\vec{v}_{\text{local}} \leftarrow \begin{pmatrix} \sin\theta\cos\phi \\ \sin\theta\sin\phi \\ \cos\theta \end{pmatrix}$$

// Orthonormal basis construction (Gram-Schmidt)
if $|d_3| < 0.999$ **then**
    $\vec{a} \leftarrow (0,0,1)$
else
    $\vec{a} \leftarrow (1,0,0)$
end if
$\vec{u} \leftarrow \frac{\vec{a}\times\vec{d}}{\|\vec{a}\times\vec{d}\|}$ // Orthogonal vector
$\vec{v} \leftarrow \vec{d} \times \vec{u}$ // Complete right-handed basis
// Rotate local vector into global coordinates
$\vec{v}_{\text{global}} \leftarrow \vec{u}(\vec{v}_{\text{local}})_x + \vec{v}(\vec{v}_{\text{local}})_y + \vec{d}(\vec{v}_{\text{local}})_z$
return $\vec{v}_{\text{global}}$

4.2.2 Ray Traversal

The ray traversal algorithm is responsible for simulating the propagation of a photon through the voxel grid. It is used to forward the photon until it reaches the first chemical compound, which is not air (or is air). It is also used to simulate the photon transport within the phantom until it reaches the exit point of the phantom.

The Input values of the algorithm are:

- The voxel grid *materialGrid* representing the geometry of the scene.
- The initial position of the photon A.
- The (unit) direction of the photon $\vec{\omega}$.

The algorithm traverses the photon through the voxel grid, voxel by voxel while checking the material of the next voxel and iterrupts in case the next voxel is not air (or is air).

The return values of the algorithm are:

- The coordinates of the final postion: C
- An array of all crossed voxel indices: *crossedVoxels*
- An array of all crossed voxel materials: *crossedMaterials*
- An array of al entry points of each crossed voxel: *entryPoints*
- An array of the distances traveres in each voxel: *distances*
- A boolean mask indicating whether the photon exited the grid: *exitGrid*

The sequence of the algorithm can be summarized as follows:

I. Initialize empty arrays for *crossedVoxels*, *crossedMaterials*, *entryPoints* and *distances*.

II. Convert the photon position A into voxel coordinates *pos*.

III. Determine the indices of the next voxel *voxelIdx* based on the *pos* and the direction $\vec{\omega}$.

IV. Set *exitGrid* to false.

V. **If** the *voxelIdx* of the next voxel is out of bounds, set *exitGrid* to true and exit the algorithm.

VI. Determine *material* of the next *voxelIdx*

VII. **If** the *material* is not air, exit the algorithm.

VIII. **While** *material* is air:

1. Append *voxelIdx* to *crossedVoxels*.
2. Append *material* to *crossedMaterials*.
3. Append *pos* to *entryPoints*.
4. Determine *maxDist* to the next voxel boundary.
5. Append *maxDist* to *distances*.

6. Update *pos* by adding $\vec{\omega} \cdot maxDist$.
7. Determine $voxelIdx$ of the next voxel based on the updated *pos*.
8. **If** the $voxelIdx$ is out of bounds, set $exitGrid$ to true and exit the algorithm.
9. Determine *material* of the next $voxelIdx$.

When the algorithm finishes, the final Position *C* is set to the last value of *position*.

To use this agorithm for travering the photons through air until they reach the phantom and to traverse the photons through the phantom until they reach the exit point of the phantom, the algorithm is called with the boolean value $throughAir$. In case $throughAir = true$, the algorithm will traverse the photons through air until they reach the first material which is not air. In case $throughAir = false$, the algorithm will traverse the photons through the phantom until they reach the exit point of the phantom.

Therefore we define a short helper function in Algorithm 3 beforehand, which generates the specific expression depending on the boolean value of $throughAir$.

Algorithm 3 Ray Traversal Helper Function

Require: Boolean $throughAir$
Require: *material*
Ensure: Boolean value
1: **if** $throughAir$ **then**
2: **return** *material* = air
3: **else**
4: **return** *material* $\neq$ air
5: **end if**

Algorithm 4 implements the process of ray traversing through the voxel grid in detail.

Algorithm 4 Ray Traversal Algorithm

Require: Boolean *throughAir*
Require: Material grid $materialGrid \in \mathbb{Z}^{X \times Y \times Z}$, voxel size $s \in \mathbb{R}^+$
Require: Initial pos $A \in \mathbb{R}^3$, direction $\vec{\omega} \in \mathbb{R}^3$
Ensure: Final pos $C \in \mathbb{R}^3$
Ensure: Arrays *crossedVoxels*, *crossedMaterials*, *entryPoints*, *distances*
Ensure: Boolean *exitGrid* indicating whether the photon exited the grid

$pos \leftarrow O/s$
$exitGrid \leftarrow$ false
$currentVoxelIdx \leftarrow \lfloor pos \rfloor$
$negDir \leftarrow \vec{\omega} < 0$
$onB \leftarrow (pos = \lfloor currentVoxelIdx \rfloor)$
$nextVoxelIdx \leftarrow currentVoxelIdx$
$nextVoxelIdx[negDir \wedge onB] \leftarrow nextVoxelIdx[negDir \wedge onB] - 1$
if any($nextVoxelIdx < 0$) or any($nextVoxelIdx \geq$ shape($materialGrid$)) **then**
    $exitGrid \leftarrow$ true
    $C \leftarrow pos$
    return $C, crossedVoxels, crossedMaterials, entryPoints, distances, exitGrid$
end if
$material \leftarrow materialGrid[nextVoxelIdx]$
while Algorithm 3(throughAir, material) **do**
    $crossedVoxels$.append($nextVoxelIdx$)
    $crossedMaterials$.append($material$)
    $entryPoints$.append(pos)
    $fracPos \leftarrow pos - \lfloor pos \rfloor$
    $maxDist \leftarrow [\infty, \infty, \infty]$
    $negDir \leftarrow \vec{\omega} < 0$
    $posDir \leftarrow \vec{\omega} > 0$
    $onB \leftarrow (pos = \lfloor currentVoxelIdx \rfloor)$
    $maxDist[negDir \wedge onB] \leftarrow -1/\vec{\omega}[negDir \wedge onB]$
    $maxDist[negDir \wedge \neg onB] \leftarrow -fracPos[negDir \wedge \neg onB]/\vec{\omega}[negDir \wedge \neg onB]$
    $maxDist[posDir] \leftarrow (1 - fracPos[posDir])/\vec{\omega}[posDir]$
    $distance = \min(maxDist)$
    $distances$.append($maxDist$)
    $pos \leftarrow pos + \vec{\omega} \cdot distance$
    $currentVoxelIdx \leftarrow \lfloor pos \rfloor$
    $nextVoxelIdx \leftarrow currentVoxelIdx$
    $nextVoxelIdx[negDir \wedge onB] \leftarrow nextVoxelIdx[negDir \wedge onB] - 1$
    if any($nextVoxelIdx < 0$) or any($nextVoxelIdx \geq$ shape($materialGrid$)) **then**
        $exitGrid \leftarrow$ true
        $C \leftarrow pos$
        return $C, crossedVoxels, crossedMaterials, entryPoints, distances, exitGrid$
    end if
    $material \leftarrow materialGrid[nextVoxelIdx]$
end while
$C \leftarrow pos$
return $C, crossedVoxels, crossedMaterials, entryPoints, distances, exitGrid$

4.2.3 Forced Detection

The forced detection algorithm is responsible to apply one iteration of the forced detection process to a photon within the phantom. Hereby the algorithm accounts:

- The voxel grid *materialGrid* representing the geometry of the scene.
- The voxel grid *totalAttenuationGrid* representing the attenuation coefficients of the materials in the voxel grid *materialGrid*.
- The voxel grid *comptonAttenuationGrid* representing the Compton scattering coefficients of the materials in the voxel grid *materialGrid*.
- The voxel grid *absorptionGrid* representing the absorption coefficients of the materials in the voxel grid *materialGrid*.
- The initial position of the photon A_i.
- The (unit) direction of the photon $\vec{\omega}_i$.
- The initial energy of the photon E_i.
- The initial intensity of the photon W_i.
- A random variable $u \sim \mathcal{U}(0,1)$ to sample the free path length.
- The voxel size s.

The algorithm then applies the ray traversal algorithm (Algorithm 4) to traverse the photon through the phantom until it reaches the exit point of the phantom C_i. Obtaining the exit point C_i, the arrays of *crossedVoxels* and *distances*, now the attenuation coefficients of the materials along the ray can be extracted from the voxel grids for the crossed voxels:

$$\begin{aligned} totalAttenuationCoefficients &= totalAttenuationGrid[crossedVoxels] \\ comptonScatteringCoefficients &= comptonAttenuationGrid[crossedVoxels] \end{aligned}$$

With the following helper algorithm (Algorithm 5), the *escapeProbability* is computed. Further the last distance index k and last interpolation factor f are returned, to solve Equation 4.3 for the free path length t_i, which can then be simply computed by:

$$t_i = \sum_{j=0}^{k-1} distances[j] + f \tag{4.6}$$

Now the next scatter point $A_{i+1} = A_i + t_i \cdot s \cdot \vec{\omega}_i$ can be determined and the new intensities together with the exit point C_i can be computed:

Algorithm 5 Partial Product Sum for Free Path Sampling in Forced Detection

Require: Arrays $distances, totalAttenuationCoefficients \in \mathbb{R}^n$, weight $u \in [0, 1]$
Ensure: Last distance index k, last interpolation factor f, $escapeProbability$

$S_{\text{total}} \leftarrow 0$
Initialize array $P[0 \dots n-1]$
for $i \leftarrow 0$ to $n-1$ **do**
    $P[i] \leftarrow distances[i] \cdot totalAttenuationCoefficients[i]$
    $S_{\text{total}} \leftarrow S_{\text{total}} + P[i]$
end for
$T \leftarrow u \cdot S_{\text{total}}$
$S \leftarrow 0$
for $i \leftarrow 0$ to $n-1$ **do**
    if $S + P[i] > T$ **then**
        $f \leftarrow \frac{T-S}{totalAttenuationCoefficients[i]}$
        return (i, f, S_{total})
    end if
    $S \leftarrow S + P[i]$
end for
return $(n, 1.0, S_{\text{total}})$ // u = 1.0 or exact fit

$$
\begin{aligned}
voxelIdx &= \lfloor A_{i+1}/s \rfloor \\
W_{i+1} &= W_i \cdot (1 - escapeProbability) \cdot \frac{comptonAttenuationGrid[voxelIdx]}{totalAttenuationCoefficients[voxelIdx]} \\
W_{i+1}^{\text{escape}} &= W_i \cdot escapeProbability
\end{aligned}
$$

And accordingly the forced detection algorithm returns the following values:

- The new position of the photon A_{i+1}.
- The new intensity of the photon after the scatter event W_{i+1}.
- The intensity of the photon for the case of escaping the phantom W_{i+1}^{escape}.
- The exit point of the phantom C_i.
- The Compton scattering coefficient μ_{CS} at the scatter point A_{i+1}.

Algorithm 6 Forced Detection Algorithm

Require: Material grid $materialGrid \in \mathbb{Z}^{X \times Y \times Z}$
Require: Total attenuation grid $totalAttenuationGrid \in \mathbb{R}^{X \times Y \times Z}$
Require: Compton scattering grid $comptonAttenuationGrid \in \mathbb{R}^{X \times Y \times Z}$
Require: Absorption grid $absorptionGrid \in \mathbb{R}^{X \times Y \times Z}$
Require: Initial position $A_i \in \mathbb{R}^3$
Require: Unit direction $\vec{\omega}_i \in \mathbb{R}^3$
Require: Initial energy $E_i \in \mathbb{R}^+$
Require: Initial intensity $W_i \in \mathbb{R}^+$
Require: Random variable $u \sim \mathcal{U}(0,1)$
Require: Voxel size $s \in \mathbb{R}^+$
Ensure: New position $A_{i+1} \in \mathbb{R}^3$
Ensure: New intensity $W_{i+1} \in \mathbb{R}^+$
Ensure: Escape intensity $W_{i+1}^{\text{escape}} \in \mathbb{R}^+$
Ensure: Exit point $C_i \in \mathbb{R}^3$
Ensure: Compton Attenuation coefficient μ_{CS}

$C_i, crossedVoxels, crossedMaterials, entryPoints, distances, exitGrid \leftarrow$ Algorithm 4$(throughAir = true, materialGrid, s, A_i)$
if exitGrid **then**
    return $(C_i, W_i, 0, C_i)$ // Photon escaped the phantom
end if
$totalAttenuationCoefficients \leftarrow totalAttenuationGrid[crossedVoxels]$
$comptonScatteringCoefficients \leftarrow comptonAttenuationGrid[crossedVoxels]$
$absorptionCoefficients \leftarrow absorptionGrid[crossedVoxels]$
$k, f, escapeProbability \leftarrow$ Algorithm 5$(distances, totalAttenuationCoefficients, u)$
$escapeProbability \leftarrow \frac{escapeProbability}{\sum_{j=0}^{k-1} distances[j] \cdot totalAttenuationCoefficients[j]}$
$t_i \leftarrow \sum_{j=0}^{k-1} distances[j] + f$
$A_{i+1} \leftarrow A_i + t_i \cdot s \cdot \vec{\omega}_i$
$voxelIdx \leftarrow \lfloor A_{i+1}/s \rfloor$
$\mu_{\text{CS}} \leftarrow comptonScatteringCoefficients[voxelIdx]$
$W_{i+1} \leftarrow W_i \cdot (1 - escapeProbability) \cdot \frac{\mu_{\text{CS}}}{totalAttenuationCoefficients[voxelIdx]}$
$W_{i+1}^{\text{escape}} \leftarrow W_i \cdot escapeProbability$
return $(A_{i+1}, W_{i+1}, W_{i+1}^{\text{escape}}, C_i, \mu_{\text{CS}})$

4.2.4 Photon Exit Point Determination

To determine the exit point of the photon in the world after exiting the phantom, a more efficient algorithm than the ray traversal algorithm from Section 4.2.2 is used can be applied. This algorithm yields the exit point of the phantom which is used to determin the detector pixel the photon contributes to.

The algorithm is initialized with the following parameters:

- The voxel grid shape (N_x, N_y, N_z) of the *materialGrid* representing the geometry of the scene.
- The initial position of the photon C^{phantom}.
- The (unit) direction of the photon $\vec{\omega}$.

- The voxel size s.

The algorithm then computes the point where the photon exits the voxel grid efficiently and returns the exit point C^{grid} and the according Coordinates $C^{\text{gridCoords}}$. The algorithm is implemented as follows:

Algorithm 7 Compute Exit Point of Ray from Voxel Grid

Require: Ray origin $C^{\text{phantom}} \in \mathbb{R}^3$, direction $\vec{\omega} \in \mathbb{R}^3$
Require: Grid shape (N_x, N_y, N_z), voxel size $s \in \mathbb{R}^+$
Ensure: Exit point C^{grid}, $C^{\text{gridCoords}}$

$C^{\text{phantomCoords}} \leftarrow C^{\text{phantom}} / s$ // Convert to voxel coordinates
$x_{\min} \leftarrow 0, x_{\max} \leftarrow N_x$
$y_{\min} \leftarrow 0, y_{\max} \leftarrow N_y$
$z_{\min} \leftarrow 0, z_{\max} \leftarrow N_z$
for axis in {x, y, z} **do**
    $o \leftarrow C^{\text{phantomCoords}}_{\text{axis}}$
    $d \leftarrow \vec{\omega}_{\text{axis}}$
    if $d > 0$ **then**
        $t^{(\text{axis})} \leftarrow \frac{x_{\max} - o}{d}$
    else if $d < 0$ **then**
        $t^{(\text{axis})} \leftarrow \frac{x_{\min} - o}{d}$
    else if $d = 0$ **then**
        $t^{(\text{axis})} \leftarrow -\infty$
    end if
end for
$t_{\text{exit}} \leftarrow \min(t^{(x)}, t^{(y)}, t^{(z)})$ // Exit time
$C^{\text{gridCoords}} \leftarrow C^{\text{phantomCoords}} + t_{\text{exit}} \cdot \vec{\omega}$
$C^{\text{grid}} \leftarrow C^{\text{gridCoords}} \cdot s \cdot \vec{\omega}$
return C^{grid}, $C^{\text{gridCoords}}$

4.2.5 Compton Scattering

The Compton scattering algorithm is responsible for simulating the physical process for the event of Compton scattering. The algorithm is initialized with the following parameters:

- The initial energy of the photon E_i.
- The (unit) direction of the photon $\vec{\omega}_i$.
- The Compton scattering attenuation coefficient μ_{CS} at the position A_{i+1}.
- Two random variables $u_1, u_2 \sim \mathcal{U}(0,1)$ to sample the new photon energy and direction.

The algorithm applies the Klein-Nishina procedure to compute the scatter angle and therefore utilizes the electron rest energy E_{rest} and follow this procedure:

I. Initialize $\kappa = E_i / E_{\text{rest}}$, $\quad \varepsilon_0 = \frac{1}{2\kappa+1}$.

II. Rejection Sampling loop:

    1. Generate random number $r \sim \mathcal{U}(0,1)$.

2. **If** $r < 0.5$, set $\varepsilon = \varepsilon_0 + (1 - \varepsilon_0) \cdot 2r$.
3. **If** $r \geq 0.5$, set $\varepsilon = \varepsilon_0 + (1 - \varepsilon_0) \cdot 2(1 - r)$.
4. Compute: $\cos\theta = 1 + \frac{1}{\kappa} \cdot (1 - \frac{1}{\varepsilon})$.
5. **If** $|\cos\theta| \leq 1$, the following is calculated:

$$\sin^2\theta = 1 - \cos^2\theta$$

If $u_1 \leq \frac{1}{2}\left(\varepsilon + \frac{1}{\varepsilon} - \sin^2\theta\right)$,

- **return** $\theta = \arccos(\cos\theta)$

III.

Given the scatter angle θ, the new photon energy E_{i+1} is computed as in [5]:

$$E_{i+1} = \frac{E_i}{1 + \kappa(1 - \cos\theta)}. \tag{4.7}$$

With the random variable u_2 an azimuthal angle ϕ is sampled to determine the new direction.

$$\phi = 2\pi u_2 \tag{4.8}$$

Assuming the vector $\vec{u}$ is an orthogonal unit vector $\vec{u} \perp \vec{\omega}_i$ and $\vec{v} = \vec{u} \times \vec{\omega}_i$, then the new direction $\vec{\omega}_{i+1}$ is as follows:

$$\vec{\omega}_{i+1} = \sin(\theta)cos(\phi) \cdot u + \sin(\theta)sin(\phi) \cdot v + \cos(\theta) \cdot \vec{\omega}_i \tag{4.9}$$

This leads to the following return values of the algorithm:

- The new photon energy E_{i+1} after the Compton scattering event.
- The new (unit) direction $\vec{\omega}_{i+1}$ of the photon after the Compton scattering event.

In the pseudoalgorithm is more explicitly describing this process in detail in Algorithm 8 by implementing omitted tweaks and details.

Algorithm 8 Compton Scattering Algorithm

Require: Initial photon energy E_i, direction $\vec{\omega}_i$
Require: Compton scattering coefficient μ_{CS} at A_{i+1}
Require: Random variables $u_1, u_2 \sim \mathcal{U}(0, 1)$
Ensure: New photon energy E_{i+1}, direction $\vec{\omega}_{i+1}$

4.3 Algorithm Composition

The main algorithm is composing the sub-algorithms from Section 4.2.

The algorithm is initialized with the following parameters:

- The voxel grid *materialGrid* representing the geometry of the scene.

- The voxel grid *totalAttenuationGrid* representing the total attenuation coefficients of the materials in the voxel grid *materialGrid*.
- The voxel grid *comptonAttenuationGrid* representing the Compton scattering coefficients of the materials in the voxel grid *materialGrid*.
- The voxel grid *absorptionGrid* representing the absorption coefficients of the materials in the voxel grid *materialGrid*.
- The source position S of the X-ray tube.
- The number of scatter events N to simulate.
- The opening angle α of the X-ray beam.
- The voxel size s of the voxel grid.
- A sequence of random variables $u \sim \mathcal{U}(0,1)^{3(N+1)}$ to sample the photon energies, directions and free path lengths.

Hereby, the photon generation algorithm is called to generate the photon energies and directions.

Then the photon is initialized with the an initial energy E_0, an initial direction $\vec{\omega}_0$ and an initial intensity $W_0 = 1$ by applying Algorithm 1 and Algorithm 2. Based on the initial position $A_0 = S$, direction $\vec{\omega}$, the voxel grid and voxel size s, the ray traversal algorithm (Algorithm 4) is applied to traverse the photon through the voxel grid until it reaches the first material which is not air. The exit point of the phantom is denoted as C_0.

The loop over the scatter events:

Later N iterations of the forced detection algorithm (Algorithm 6) are applied together with one random variable to simulate the photon transport through the phantom. In each iteration, the photon position and intensity are updated. With Algorithm 7, the exit point C_i^{grid} of the photon is computed and captured together with the relevant intensity $E_i \cdot W_{i+1}^{\text{escape}}$.

Then the Compton scatter event is simulated, taking into account the photon energy E_i, the direction $\vec{\omega}_i$ and the Compton scattering coefficient μ_{CS} at A_{i+1}. Together with two random variables, the new photon energy E_{i+1} with an according direction $\vec{\omega}_{i+1}$ is sampled.

Appendix A

Frequently Asked Questions

A.1 How do I change the colors of links?

The color of links can be changed to your liking using:

`\hypersetup{urlcolor=red}`, or

`\hypersetup{citecolor=green}`, or

`\hypersetup{allcolor=blue}`.

If you want to completely hide the links, you can use:

`\hypersetup{allcolors=.}`, or even better:

`\hypersetup{hidelinks}`.

If you want to have obvious links in the PDF but not the printed text, use:

`\hypersetup{colorlinks=false}`.

Bibliography

[1] H.W.A.M. de Jong, E.T.P. Slijpen, and F.J. Beekman. "Acceleration of Monte Carlo SPECT simulation using convolution-based forced detection". In: *IEEE Transactions on Nuclear Science* 48.1 (2001), pp. 58–64. DOI: 10.1109/23.910833.

[2] Guiyuan Lin, Shiwo Deng, and Xiaoqun Wang. "An efficient quasi-Monte Carlo method with forced fixed detection for photon scatter simulation in CT". In: *PLOS ONE* 18 (Aug. 2023), e0290266. DOI: 10.1371/journal.pone.0290266.

[3] *Medical Imaging Systems : An Introductory Guide*. eng. Image Processing, Computer Vision, Pattern Recognition, and Graphics; 11111. Cham, 2018. Chap. 7,8. ISBN: 9783319965208.

[4] Thomas Müller-Gronbach, Erich Novak, and Klaus Ritter. *Monte Carlo-Algorithmen*. Springer-Verlag, 2012.

[5] Dylan R Nelson. "COMPTON SCATTERING". In: (2007). URL: https://www.ita.uni-heidelberg.de/~dnelson/storage/ucb.phy111.spr2007/dnelson.com.pdf.

[6] Gavin Poludniowski. *SpekPy: Python package for simulating X-ray spectra*. Version 2.0.13. 8.08.2024. URL: https://bitbucket.org/caxtus/book/src/master/.

[7] Gavin Poludniowski, Artur Omar, and Pedro Andreo. *Calculating X-ray Tube Spectra: Analytical and Monte Carlo Approaches*. CRC Press, 2022.

[8] Gavin Poludniowski et al. "SpekPy v2. 0—a software toolkit for modeling x-ray tube spectra". In: *Medical Physics* 48.7 (2021), pp. 3630–3637.

[9] A. Sisniega et al. "Automatic Monte-Carlo Based Scatter Correction For X-ray cone-beam CT using general purpose graphic processing units (GP-GPU): A feasibility study". In: *2011 IEEE Nuclear Science Symposium Conference Record*. 2011, pp. 3705–3709. DOI: 10.1109/NSSMIC.2011.6153699.

[10] Jörg Steidel et al. "Dose reduction potential in diagnostic single energy CT through patient-specific prefilters and a wider range of tube voltages". In: *Medical physics* 49.1 (2022), pp. 93–106.

[11] Murugesan Venkatapathi et al. "An O (n) algorithm for generating uniform random vectors in n-dimensional cones". In: *arXiv preprint arXiv:2101.00936* (2021).