Università degli Studi di Pavia

Dipartimento di Fisica

Corso di Laurea Magistrale in Fisica

MC Simulation of Cone Beam CT using the ICRP 110 Phantom: an head and neck example.

Simulazione MC di TC a fascio conico usando il fantoccio ICRP 110: esempio del complesso testa-collo.

MSc thesis by:
Marco Paganelli

Supervisor:
Prof. Ian Postuma
Co-supervisors:
Dr. Setareh Fatemi
Dr. Daniele Pistone

A.A. 2023-2024

"fatti non foste a viver come bruti,
ma per seguir virtute e canoscenza."
Dante, Divina Commedia, Inf. XXVI 119-120

Abstract

Questa tesi esplora l'uso della simulazione Monte Carlo (MC) per generare immagini di Tomografia Computerizzata (CT) utilizzando il fantoccio ICRP110, con particolare attenzione alla regione testa-collo. Studiare e riprodurre dati volumetrici in questa regione è utile nel contesto della Boron Neutron Capture Therapy (BNCT), poichè uno dei principali target terapeutici della BNCT sono i tumori testa collo. La BNCT rappresenta una promettente alternativa alle terapie convenzionali per i tumori della testa e del collo, grazie alla sua capacità di danneggiare in maniera importante le cellule tumorali riducendo il danno ai tessuti sani circostanti.

La generazione di immagini CT *in-silico* da fantocci umanoidi è di particolare interesse nel contesto dell'intelligenza artificiale. Generalmente i dataset disponibili per allenare modelli di *deep learning* sono composti da pochi dati che raramente contengono pazienti sani. Avere un sistema che permetta di generare immagini diagnostiche volumetriche ci permetterebbe non solo di ampliare i dataset ma anche di migliorare l'allenamento degli algoritmi. In questo contesto, a Pavia è stato sviluppato il progetto AI_MIGHT il quale ha come fine allenare modelli di deep learning per contornare volumi tumorali in modo automatico.

In questa tesi viene sviluppato un primo approccio di simulazione MC per la generazione di immagini CT *in-silico*. Per poter raggiungere questo obiettivo abbiamo utilizzando i codici GATE e Geant4, cercando di riprodurre l'acquisizione di un immagine diagnostica da fascio conico (Cone Beam CT (CBCT)). Il lavoro descrive il processo di ricostruzione volumetrica di un CBCT, ottenuto simulando l'interazione di un fascio conico di fotoni con il fantoccio ICRP 110, e la successiva elaborazione delle immagini tramite il programma OSCaR.

I risultati ottenuti dimostrano come sia possibile creare immagini CBCT generate dal fantoccio ICRP110 attraverso metodi MC. Tuttavia, queste immagini non possono essere utilizzate per addestrare sistemi di intelligenza artificiale. Ciò nonostante in questo lavoro è stata impostata la simulazione Geant4. Affinando questa, sarà possibile implementare sorgenti e rivelatori per generare i dati necessari alla ricostruzione di una immagine CT.

Le prospettive future includono l'ottimizzazione della simulazione, a partire dalla definizione della sorgente arrivando all'implementazione di rivelatori più complessi. Oltre alla simulazione sarà necessario implementare tecniche di ricostruzione dell'immagine alternative. Ad esempio avere un movimento ad elica della sorgente per migliorare la qualità

dell'immagine ricostruita. Inoltre, per aumentare il numero di immagini da inserire nel dataset per allenare algoritmi di intelligenza artificiale si possono usare altri tipi di fantoccio e modellarli a seconda delle necessità.

Questo lavoro fa parte del progetto AI_MIGHT dedicato all'implementazione di software per la segmentazione automatica di immagini mediche per l'ottimizzazione del trattamento BNCT.

Abstract

This thesis explores the use of Monte Carlo (MC) simulation to generate Computed Tomography (CT) images using the ICRP110 phantom, with a focus on the head-neck region. Studying and reproducing volumetric data in this region is useful in the context of Boron Neutron Capture Therapy (BNCT), as one of the main therapeutic targets of BNCT are head-neck tumours. BNCT represents a promising alternative to conventional therapies for head and neck cancers due to its ability to significantly damage tumour cells while reducing damage to surrounding healthy tissue.

The generation of *in-silico* CT images from humanoid phantoms is of particular interest in the context of artificial intelligence. Generally, the datasets available to train *deep learning* models are composed of few data that rarely contain healthy patients. Having a system that enables the generation of volumetric diagnostic images would allow us not only to expand the datasets but also to improve the training of the algorithms. In this context, the AI_MIGHT project was developed in Pavia, which aims to train deep learning models to contour tumour volumes automatically.

In this thesis, a first MC simulation approach for the generation of *in-silico* CT images is developed. In order to achieve this goal, GATE and Geant4 codes has been used, attempting to reproduce the acquisition of a diagnostic Cone Beam image (Cone Beam CT (CBCT)). The work describes the process of volumetric reconstruction of a CBCT, obtained by simulating the interaction of a cone beam of photons with the ICRP 110 phantom, and the subsequent image processing using the OSCaR software.

The results obtained demonstrate that it is possible to create CBCT images generated by the ICRP110 phantom through MC methods. However, these images cannot be used to train artificial intelligence systems. Nevertheless, the Geant4 simulation was set up in this work. By refining this, it will be possible to implement sources and detectors to generate the data required for the reconstruction of a CT image.

Future perspectives include the optimisation of the simulation, starting with the definition of the source and ending with the implementation of more complex detectors. In addition to simulation, it will be necessary to implement alternative image reconstruction techniques. For example, having a helical movement of the source to improve the quality of the reconstructed image. Furthermore, to increase the number of images to be included in the dataset for training artificial intelligence algorithms, other types of phantom can be used and modelled as required.

This work is part of the AI_MIGHT project dedicated to the implementation of software for the automatic segmentation of medical images for the optimisation of BNCT treatment.

Contents

Introduction

Head and neck cancer is a general term encompassing multiple cancers that can develop in the head and neck region include cancers of the mouth, tongue, gums and lips (oral cancer), voice box (laryngeal), throat (nasopharyngeal, oropharyngeal, hypopharyngeal), salivary glands, nose and sinuses[1]. About 90% are pathologically classified as squamous cell cancers[2]. In 2018, it was the seventh most common cancer worldwide, with 890,000 new cases documented and 450,000 people dying from the disease[3]. Head and neck cancers (HNCs), particularly squamous cell carcinomas (SCCs), are among the most challenging types of cancers to treat due to their anatomical complexity and the vital structures involved. Conventional therapies, including surgery, radiation therapy, and chemotherapy, often have limited success, especially in cases of locally advanced or recurrent disease. Boron Neutron Capture Therapy (BNCT) presents a promising alternative due to its ability to deliver a highly localized cytotoxic effect while sparing normal tissues [4]. Clinical trials of BNCT for head and neck cancers have shown encouraging results [5].

Treatment Planning Systems (TPS) are crucial tools in radiation therapy, used for the calculation and optimization of the dose distribution within the patient, accurate delineation of target volumes is essential for the BNCT success, and it directly impacts on the treatment effectiveness. This process is known as tumor contouring. Artificial Neural Network (ANN) has provided new methods of automatic tumor contouring, potentially capable to enhance and speed up the processing of an adequate treatment plan.

To perform clinical treatments in a radiation therapy set-up, treatment planning simulations are carried out to evaluate the clinical outcome. For this reason, contouring the tumour volume and the organs at risk is an important task. This process involves defining the region of interest boundaries, a task which is time-consuming and tedious. For this reason, artificial neural networks (ANNs) are more and more used in diagnostic medical imaging. However, training a high-performing neural network requires a large dataset to a well generalizing model. In the medical field, contoured imaging datasets are often quite limited in size. Moreover, these datasets usually contain patients who already have a diagnosed disease, meaning that the neural networks are primarily exposed to abnormal tissue patterns, such as tumors. This creates a bias in training because the ANN lacks exposure to healthy tissue, which is essential for distinguishing between normal and pathological anatomy. This work wants to explore the generation of healthy tissue data from Monte Carlo simulation.

The aim of this thesis is to generate Cone Beam CT images of healthy patients in order to increase the dataset from which AI can be trained for treatment planning. As such, this works is sinergic with the "AI_MIGHT" project (Artificial Intelligence methods applied to Medical ImaGes to enHance and personalize BNCT Treatment planning) which aims to apply deep learning techniques in BNCT. In particular it woul help to obtain a healthy patient model to segment healthy tissues in an automatic way.

The work focused on the volumetric reconstruction of a CBCT scan of a ICRP110 phantom. In particular the CBCT setup was simulated using Geant4 Monte Carlo code.

In the first chapter, the principles of BNCT are discussed with focus on the physical process on which the therapy is based and on the selectivity in this technique. One of the targets of BNCT are tumours of the head and neck distrect,therefore, this work hase been focus on such region.

In the second chapter an introduction to the bssic concepts of Computed Tomography and Radiographic Images are introduced. In particular, the chapter focuses on the CBCT and the image reconstruction methods for this imaging modality.

The third chapter of this thesis gives an introduction to the Monte Carlo methods that were susbequentelly employed for the thesis objective. Specifically the GATE toolkit and Geant4 Monte Carlo code are introduced.

In the fourth chapter the code to recreate the CBCT reconstruction will be delved into, starting from the first simple example performed on GATE of a microCBCT. From this first example the tools needed for the project has been developed.Afterwards the focus moved onto Geant4 and the various section of the code will be explained. Using Geant4 it was possible to obtain the different projections of the CBCT of the ICRP110phantom that were than reconstructed trough the OSCaR software.

In the last two chapter the results obtained of the CBCT of ICRP110 phantom are presented and is explained how they can be improved to be used in the AI dataset, and the future perspectives of this work.

Chapter 1

In-silico computed tomography for Head and Neck region in BNCT

1.1 Principles of Boron Neutron Capture Therapy (BNCT)

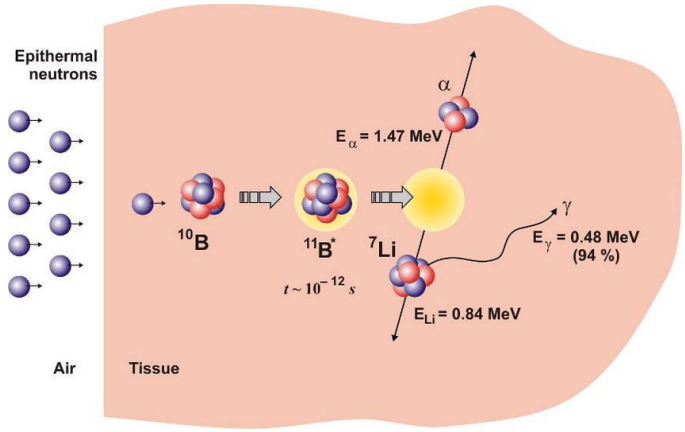


Figure 1.1: Nuclear reaction in BNCT.[6]

Boron Neutron Capture Therapy (BNCT) is an experimental form of radiotherapy that uses the properties of the neutron capure reaction on the boron isotope ^{10}B. Which can be used to selectively target and destroy cancer cells while minimizing damage to surrounding healthy tissues. The principle behind BNCT is relatively straightforward. If boron atoms can be preferentially accumulated in tumor cells and the area is subsequently exposed to a neutron beam, boron neutron capture reactions may occur. This nuclear reaction produces two high LET particles: an alpha particles (α) and lithium nuclei (^{7}Li). These particles have a very short range in biological tissues, respectively approximately 10 and 5 micrometers. Which ensures that the energy deposited by these charged particles is confined to the boron-loaded cancer cells, thus sparing adjacent normal tissues [5].

The reaction that occurs during BNCT as shown in figure 1.1 is as follows[7]:

$$^{10}\mathrm{B} + n \rightarrow [^{11}\mathrm{B}]^{*} \rightarrow \alpha + ^{7}\mathrm{Li} + (2.31\ \mathrm{MeV})$$

In approximately 93.9% of cases, an excited state of lithium is produced, which decays, emitting a 478 keV gamma photon. The remaining 6.1% of reactions directly produce α and ^{7}Li without gamma emission. These particles are highly ionizing and capable of causing irreparable damage to the cell DNA, such as clustered double strain breaks. Effectively killing the cells containing boron atoms [8].

1.1.1 Mechanism of Action and Selectivity of BNCT

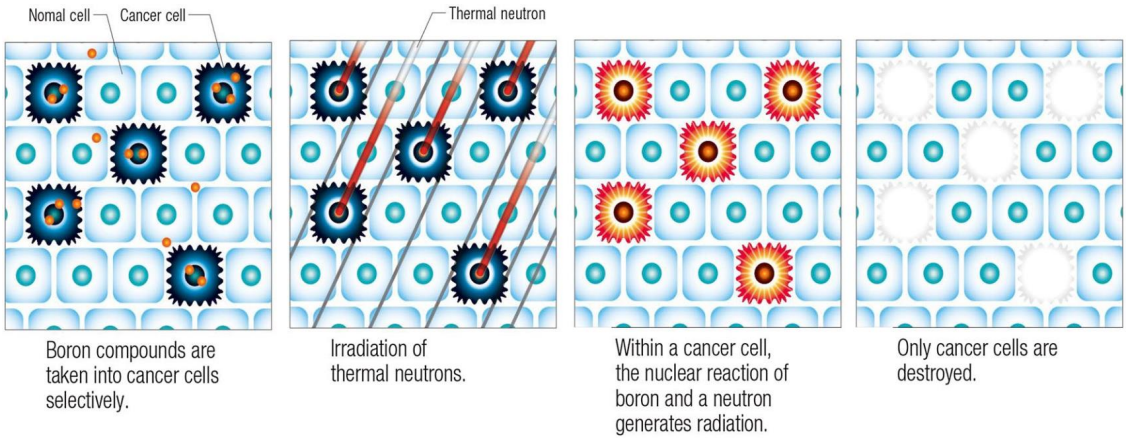


Figure 1.2: The concept of selectivity of BNCT. 1) A boron-containing drug is administered to the patient and it selectively accumulates in cancer cells. 2) The target is irradiated with a low-energy neutron beam, and the neutron captures occur in boron. 3) The thermal neutron capture by ^{10}B releases an α particle and a ^{7}Li nucleus in the cancer cell. 4) Tumor cells absorb a lethal dose while the healthy cells are spared. [9]

The therapeutic selectivity, as seen in figure 1.2, of BNCT is fundamentally different from traditional radiotherapy and hadrontherapy. While traditional methods rely on physical targeting using focused beams, BNCT's selectivity is biological; it exploits the differential uptake of boron-10 by tumor cells compared to normal cells. Tumor cells are targeted by boronated compounds, such as boronophenylalanine (BPA) and borocaptate sodium (BSH)[10][11][12], which preferentially accumulate within the malignant cells due to differences in metabolic activity as seen through FBPA-PET of figure 1.5 or cellular transport mechanisms.

Upon neutron irradiation, the boron-10 in the tumor cells captures low energy (<.5eV) thermal neutrons, leading to the release of (α particle and ^{7}Li) in recoil, causing direct highly localized DNA damage and cell death. The range of these particles is limited to the scale of a single cell, minimizing the damage to adjacent normal tissues.

The neutron capture cross section of ^{10}B is particularly significant at low neutron energies, where it dominates over other interaction types. This dominance is especially

pronounced at thermal energies, around 25 meV, where the capture cross section reaches approximately 4000 barns as can be seen in figure 1.3.

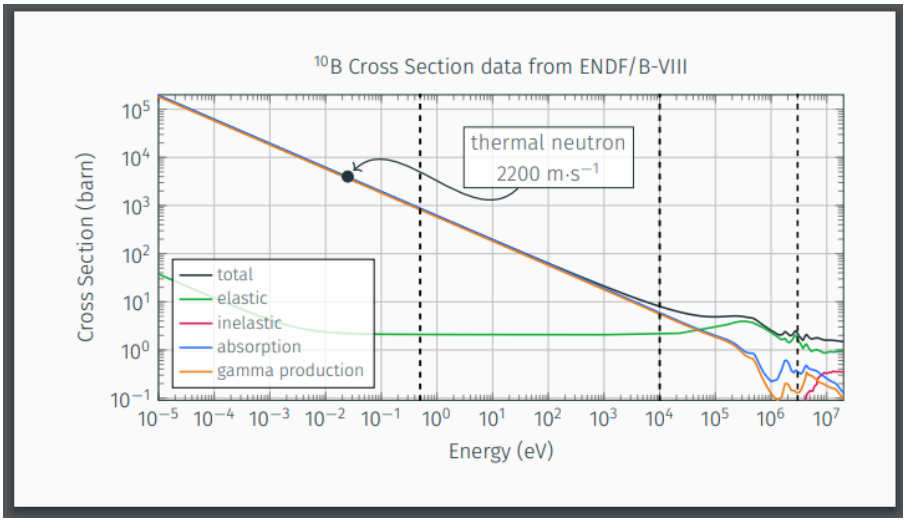


Figure 1.3: Cross sections for different interaction types with ^{10}B as a function of neutron energy. The capture cross section is clearly dominant at low energies.

This makes ^{10}B an excellent neutron absorber, particularly in thermal neutron environments.

If it is compared with the cross section of other common elements like hydrogen (^{1}H), oxygen (^{16}O), carbon (^{12}C), and nitrogen (^{14}N).

From this two plot of figure 1.4, it is evident that ^{10}B has a significantly higher capture cross section at low energies compared to hydrogen, oxygen, carbon, and nitrogen, making the selectivity of BNCT.

1.1.2 BNCT Clinical Implementation Dosimetry and Treatment Planning

The dosimetry of BNCT is more complex than conventional radiotherapy due to the mixed radiation field produced by neutron capture and secondary interactions. Dosimetric calculations must account for the contributions from boron capture reactions, proton recoil from neutron scattering, and other secondary particles generated in tissues. Monte Carlo simulations are commonly used to model radiation transport and optimize treatment plans, ensuring maximal tumor dose while minimizing exposure to healthy tissues [14]. Recent advancements in dosimetric techniques and radiobiological modeling have improved the precision of BNCT treatment planning, allowing for better prediction of clinical outcomes and reduced side effects. First the development of photon-equivalent dose models (i.e. the Coderre & Morris model) and later the photon isoeffective dose model by Gonzalez and Santa Cruz, have further refined the translation of BNCT dosimetry into clinically relevant metrics [15].

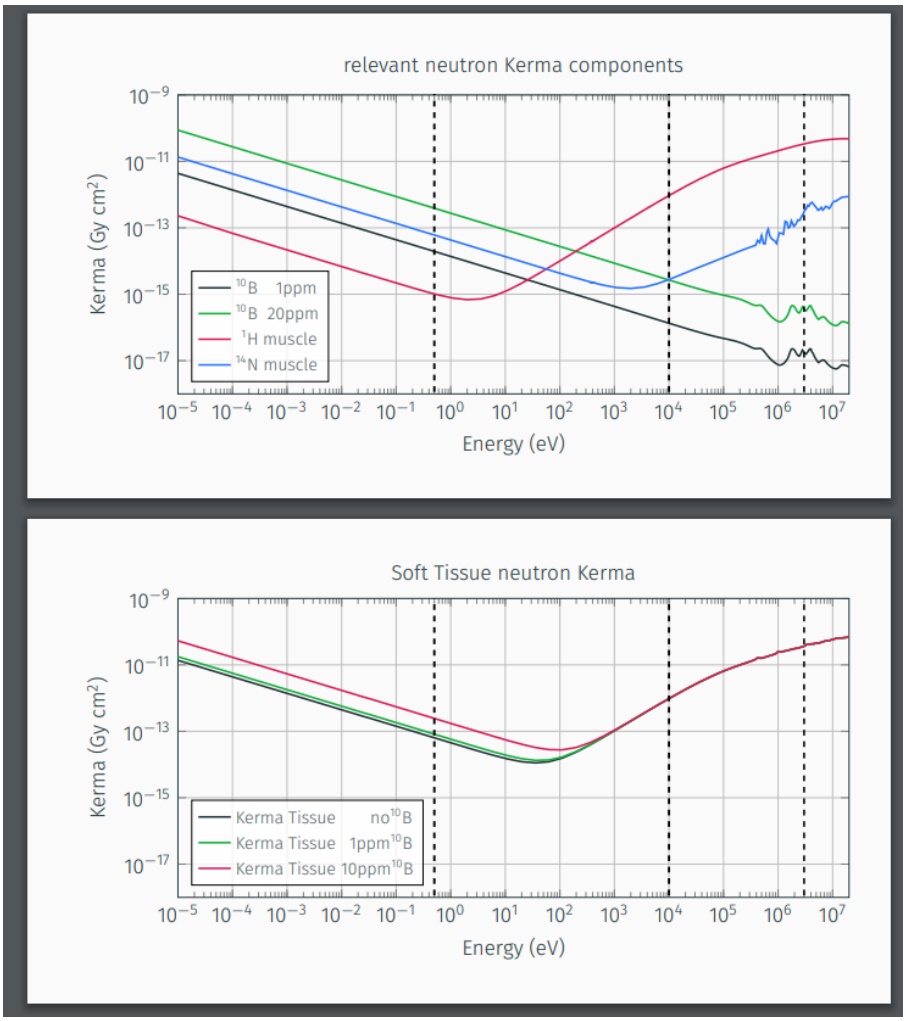


Figure 1.4: In the upper plot: Comparison of neutron Kerma for ^{10}B 1 and 20 ppm with ^{1}H and ^{14}N as a function of neutron energy; in the lower one Comparison of neutron Kerma for soft tissue with ^{10}B 1, 10 and 0 ppm.

The clinical adoption of BNCT is expanding, particularly in regions with advanced accelerator technologies and supportive regulatory environments. Japan remains at the forefront, with several facilities approved for clinical use and covered by national health insurance for the treatment of unresectable, locally advanced, and recurrent head and neck cancers. Ongoing clinical trials and technological advancements are likely to further solidify BNCT's role in the oncological landscape, particularly for tumors that are challenging to treat with conventional methods [16].
Recent research focuses on developing novel boron compounds with improved tumor selectivity, faster clearance from healthy tissues, and higher retention in tumors. These efforts aim to enhance the therapeutic efficacy of BNCT, potentially expanding its applicability to a broader range of cancers, including those with diffuse or metastatic spread.[17]

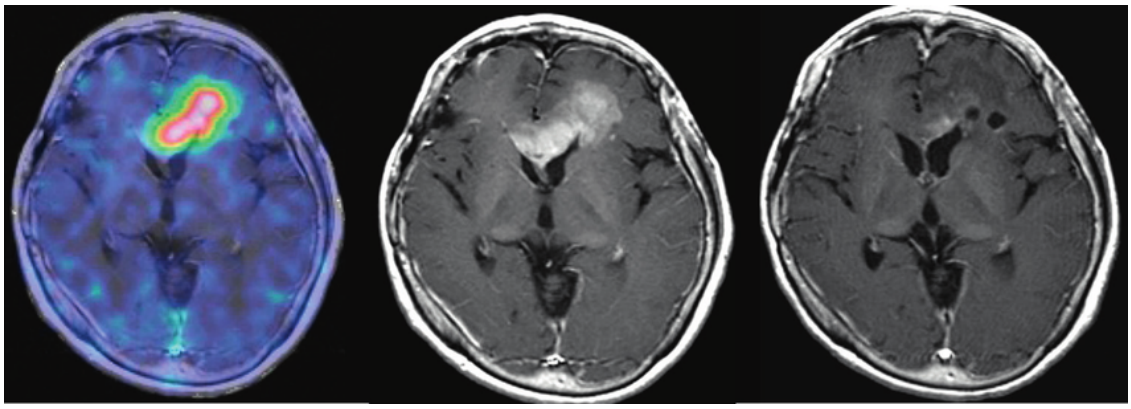


Figure 1.5: A case of malignant glioma treated with BNCT that had a significant response. In this case, favorable boron drug distribution was observed on the pretreatment PET images. (left: FBPA-PET fusion, middle: Before treatment, right: After BNCT of contrast-enhanced MRI T1WI). [13]

1.1.3 Head and Neck Cancers

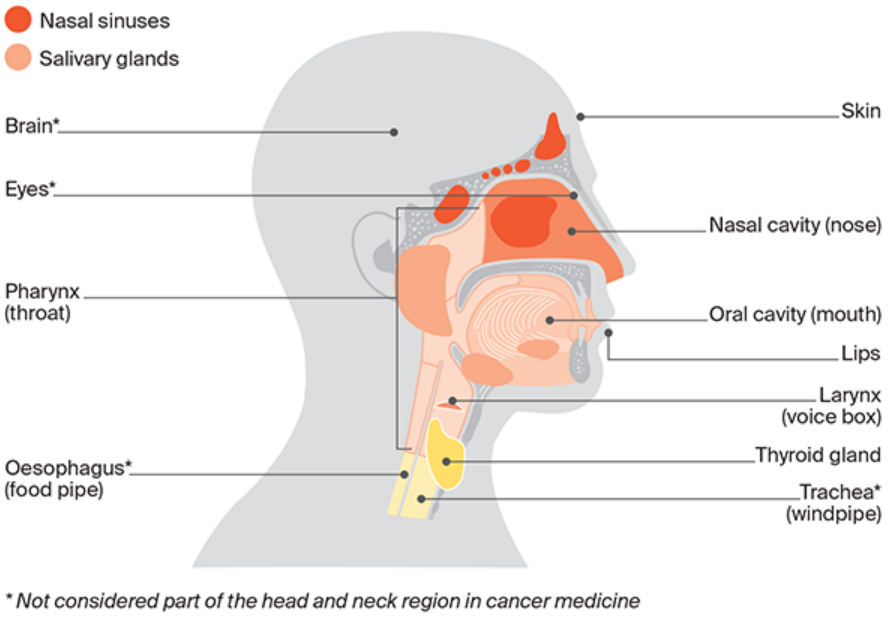


Figure 1.6: This diagram gives an overview of the main areas of the head and neck.[18]

Head and neck cancer is a general term encompassing multiple cancers that can develop in the head and neck region shematized in figure 1.6. These include cancers of the mouth, tongue, gums and lips (oral cancer), voice box (laryngeal), throat (nasopharyngeal, oropharyngeal, hypopharyngeal), salivary glands, nose and sinuses[1]. About 90% are pathologically classified as squamous cell cancers[2].Globally, head and neck cancer accounts for 650,000 new cases of cancer and 330,000 deaths annually on average. In

2018, it was the seventh most common cancer worldwide, with 890,000 new cases documented and 450,000 people dying from the disease[3]. Smoking and alcohol consumption remain two of the main risk factors for head and neck cancer, particularly regarding the oral cavity, larynx, oropharynx, and hypopharynx. These factors tend to primarily affect older individuals who have abused tobacco and alcohol throughout their lives. However, thanks to the gradual reduction in smoking habits over the past decade, there has also been a slight decline in new cases of head and neck cancers[19]. An exception to this trend concerns oropharyngeal cancers, whose incidence has increased mainly due to the human papillomavirus (HPV) type 16. In the 2000s, over 73% of oropharyngeal cancer cases in the United States were found to be HPV-positive. This virus is, in fact, one of the main causes of oropharyngeal cancers, primarily affecting younger individuals, particularly in North America and Northern Europe, with a latency that can range from 10 to 30 years after exposure to unprotected oral sex. This has led to an increase in cancers of the tonsils and the base of the tongue[20][21][22]. Other recognized risk factors include exposure to ionizing radiation, the use of chemicals in heavy industry, and poor oral hygiene[23].
Head and neck cancers (HNCs), particularly squamous cell carcinomas (SCCs), are among the most challenging types of cancers to treat due to their anatomical complexity and the vital structures involved. Conventional therapies, including surgery, radiation therapy, and chemotherapy, often have limited success, especially in cases of locally advanced or recurrent disease. BNCT presents a promising alternative due to its ability to deliver a highly localized cytotoxic effect while sparing normal tissues [4]. Effective BNCT requires boron delivery agents that selectively concentrate in tumor cells while minimizing uptake by normal tissues. The two primary boron carriers currently approved for clinical use are BPA and BSH. BPA, an amino acid analogue, is preferentially taken up by tumor cells due to enhanced amino acid transport mechanisms, while BSH relies on passive diffusion, particularly effective in targeting brain tumors across the disrupted blood-brain barrier [24].
Globally, head and neck squamous cell carcinoma (HNSCC) accounts for more than 700,000 cases and 350,000 deaths annually. Most patients present with locally advanced disease (Stages III and IV), and despite aggressive multimodality treatment approaches, recurrence rates are high (25–60%) and prognosis remains poor, with a median overall survival of less than one year for recurrent or metastatic cases [25]. The high recurrence rates and the anatomical difficulties in achieving clean surgical margins highlight the need for more targeted and effective therapeutic strategies.

Non-squamous cell carcinoma (nSCC) types, such as adenoid cystic carcinoma, are even rarer and often more resistant to conventional treatments like photon therapy and chemotherapy. BNCT has been shown to provide a higher therapeutic efficacy and safety profile for these patients, particularly those who have already undergone prior irradiation [25].

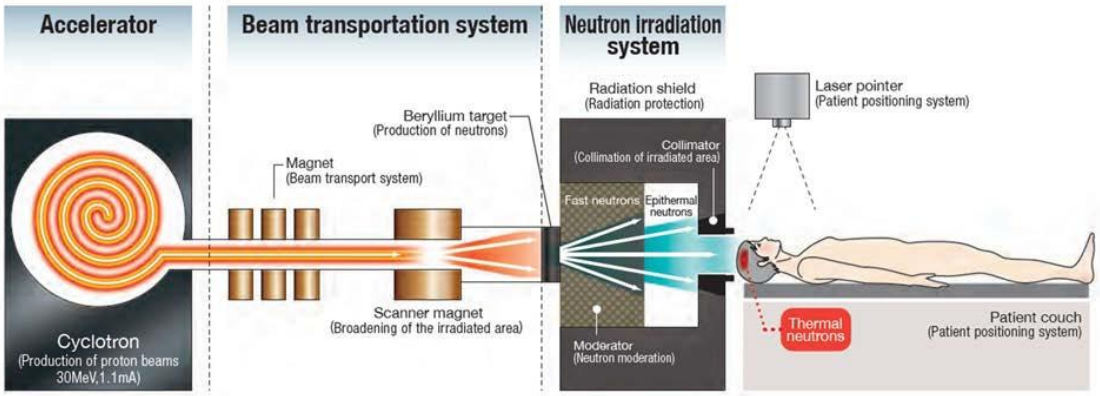


Figure 1.7: Schematic diagram of an AB-BNCT system. [26]

Clinical Studies of BNCT in Head and Neck Cancers

Clinical trials of BNCT for head and neck cancers have shown encouraging results. The first documented use of BNCT for a head and neck tumor was at Kyoto University Research Reactor Institute in 2001, treating a patient with recurrent parotid gland tumor. This initial success led to a series of trials in Japan, Finland, and Taiwan, where different boron carriers (such as BPA and BSH) and neutron sources (research reactors and accelerators) were utilized [5]. In 2001, Japanese researchers treated with BNCT a recurrent parotid gland tumor, originated from a primary tumor treated with conventional therapies, at the Kyoto University Research Reactor Institute (KURRI). It was the first treatment of this kind in the world and the result was encouraging; this success stimulated new BNCT clinical trials for H&N cancer presented below:

Table 1.1: BNCT Clinical Trials for Recurrent or Untreated Head and Neck Cancer

Institution	**Treatment Dates**	**Tumor Type, No. Patients**	**Boron Carrier and Administration**	**Outcome**	**Refs**
Research Reactors					
Kyoto University Research Reactor Institute, Japan	2001–2007	rH&N, 49; urH&N, 13	BSH+BPA (13 cases); BPA (72 cases); 250 mg/kg (5 cases); 500 mg/kg; 200 mg/kg/h × 2 h + 100 mg/kg/h × 1h during irradiation (67 cases)	PR: 28%, CR: 29%, MeST: 10.1 mos., 2y OS: 24.2%	[27]
Helsinki University Hospital, Finland	2003–2012	79	BPA 350–400 mg/kg in 2 h before irradiation	PR: 32%, CR: 36%, 2y LRPFS: 38%, 2y OS: 21%	[28]
Taipei, Veterans General Hospital, Taiwan	2010–2013	SCC: 11; nSCC: 6	BPA 450 mg/kg (180 mg kg^{-1} h^{-1} × 2h; 90 mg kg^{-1} h^{-1} × 0.5h during irradiation)	PR: 35%, CR: 35%, 2y LRPFS: 28%, 2y OS: 47%	[29] [30] [31]
Accelerators Figure 1.7					
Southern Tohoku General Hospital, Japan	2016–2019	SCC: 13; nSCC: 8	BPA 500 mg/kg (200 mg kg^{-1} h^{-1} × 2h; 100 mg kg^{-1} h^{-1} during irradiation)	PR: 48%, CR: 24%, 2y LRPFS: 28%, 2y OS: 85.3%	[32]

In the table 1.1: rH&N recurrent head and neck cancer, ur unresectable, rSqCC recurrent squamous cell carcinoma, rnSqCC recurrent non-squamous cell carcinoma, nSCC non-squamous cellcarcinoma, PR partial response, CR complete response, 2 y OS 2-year overall survival, MeST median survival time, given in months. PR partial response, CR complete response, 2 y OS 2-year overall survival, MeST median survival time, given in months.

Treatment Planning Systems (TPS) are crucial tools in radiation therapy, used for the calculation and optimization of the dose distribution within the patient. These systems allow for the development of individualized treatment protocols, taking into account factors such as beam energy, direction, field size, and fluence. The goal is to deliver the maximum dose to the target while minimizing the risk of complications to surrounding healthy tissues [33].

In clinical settings, the entire process of treatment planning is supervised by a medical physicist. Their responsibility is to ensure the accuracy and reliability of dose distribution calculations. Once the plan is complete, it is reviewed and approved by a radiation oncologist, who verifies its precision and suitability before implementation [34].

The treatment planning process generally follows these steps [35]:

- Identifying the tumor's shape and location (target) as well as nearby organs at risk using modern imaging techniques such as CT, MRI, or SPECT.

- Selecting appropriate methods for patient positioning and immobilization to ensure reproducibility during treatments.
- Optimizing and selecting a suitable beam configuration.
- Evaluating the resulting dose distribution in the targeted volumes.
- Calculating the machine settings necessary to deliver the prescribed dose.

Radiation therapies are localized treatments, meaning their anti-cancer effects are confined to the irradiated organs and tissues. Therefore, the precise definition of target volumes, adjacent organs at risk, and other anatomical structures is critical for developing accurate treatment plans and serves as a basis for comparing treatment outcomes[36].

Advancements in medical imaging techniques, along with their integration into the planning processes, will play a significant role in improving the accuracy of delineating both the Gross Tumor Volume (GTV) and Clinical Target Volume (CTV). Furthermore, enhanced capabilities to track internal organ movements during treatment will lead to better coverage of the Internal Target Volume (ITV) and, consequently, the Planning Target Volume (PTV).

The Computer Tomography scan is the golden standard of diagnostic imaging for radiotherapy and it is used to extract the Region Of Interest (ROI) segmentation.

1.2 Cone Beam Computed Tomography

This chapter is about how to carry out the CBCT reconstruction, therefore focusing on the head and neck area.The chapter will start from the generation of X-rays, and the creation of the radiographic image, and then move on to tomography and the actual reconstruction of the CBCT by introducing the OSCaR software

1.2.1 Computed Tomography (CT)

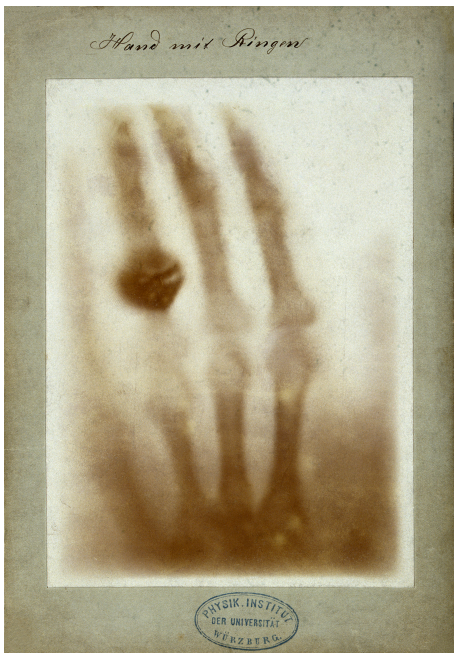


Figure 1.8: Hand mit Ringen (Hand with Rings): a print of one of the first X-rays by Wilhelm Röntgen (1845–1923) of the left hand of his wife Anna Bertha Ludwig. It was presented to Professor Ludwig Zehnder of the Physik Institut, University of Freiburg, on 1 January 1896.

Before 1896, there were no methods available to explore or measure the hidden internal structures of the living human body. Roentgen's discovery of the penetrating X-ray, in 1896, with the first image of his wife's hand as shown Fig. 1.8 started a revolution in medical imaging and began a slow process of reunification of medical science with physics, chemistry and engineering [37].

1.2.2 X-rays

The generation of X-rays needed for imaging purposes typically involves the use of an X-ray tube device that converts electrical energy into X-ray radiation.This machine consists of several key components schematized in Fig. 1.9 [39].

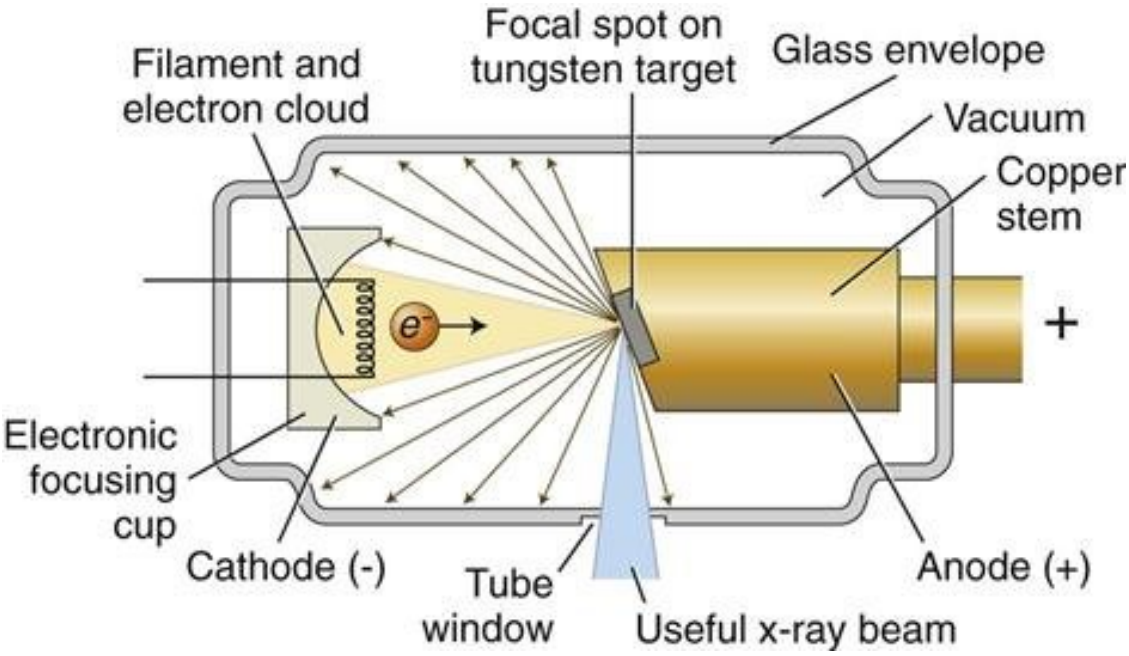


Figure 1.9: Schematics of a conventional X-ray tube. [38]

The X-ray tube operates as a highly specialized vacuum tube designed to generate X-rays by accelerating electrons from the cathode to the anode at high velocities. The cathode consists of a heated tungsten filament, typically about 0.2 mm in diameter, which emits electrons through thermionic emission when heated to approximately 2200 $°C$. The filament is coiled into a spiral of about 1 cm, creating a focused source of electron emission. Surrounding the filament is the focusing cup, often made of nickel or another conductive material, designed to direct the emitted electrons toward the anode. The focusing cup is either at the same potential as the filament or, in modern tubes, slightly negative relative to it, helping to concentrate the electron beam into a narrow stream directed at the anode.

When a high potential difference (typically in the range of tens to hundreds of kilovolts) is applied between the cathode and the anode, the electrons are rapidly accelerated across the vacuum space toward the anode. Thus, the space charge effect occurs when the applied voltage is insufficient to pull all emitted electrons from the vicinity of the filament, resulting in a saturation of the tube current. Below this saturation point, increases in applied voltage result in a higher tube current due to the space charge effect, but beyond the saturation point, all available electrons are pulled away, and the tube current becomes independent of further voltage increases, being solely determined by the filament's temperature and emission.

The anode, usually made of a high atomic number material like tungsten, serves as the target for the high-velocity electrons. When the electrons strike the anode, their kinetic energy is converted into two types of radiation: Bremsstrahlung (braking radiation) and

characteristic X-rays. Bremsstrahlung is the primary mechanism in most X-ray tubes, where the deceleration of electrons in the electric field of the atomic nuclei produces a broad spectrum of X-rays. A small fraction of the X-rays produced are characteristic of the anode material, depending on the energy levels of electrons ejected from the inner shells of the atoms.

The design of the anode varies depending on the tube's purpose. In stationary anode tubes, the anode is fixed and requires efficient cooling to dissipate the heat generated during electron collisions. In rotating anode tubes, the anode spins at high speeds to distribute the heat over a larger surface area, enabling higher tube currents and shorter exposure times.

The focusing cup, in modern tubes, can serve a dual purpose by acting as a grid. When biased with a negative potential relative to the filament, it controls the flow of electrons by repelling them and effectively "pinching off" the tube current, allowing for rapid control of the X-ray production without the need to switch the high voltage on and off.

The tube is housed within a protective casing filled with insulating oil or another medium to absorb excess heat and ensure electrical insulation. A window, typically made of plastic, allows the X-rays to exit the tube with minimal attenuation.

The tube is made of Pyrex glass under vacuum that allows to control independently the number and the speed of electrons. The X-rays are emitted isotropically, thus needing a shield of lead all around it to prevent the radiation to exceed $100mR$ in 1 hour at the maximum current when at 1 m from the source. A plastic window is placed in the shielding for the X-rays to emerge. X-rays are a form of electromagnetic radiation with wavelengths ranging from about 0.1 to 100 keV. In an X-ray tube, the production of X-rays occurs primarily through two mechanisms: Bremsstrahlung Radiation (Braking Radiation) or Characteristic X-Ray Emission the spectrum can be seen in Fig. 1.10.

Bremsstrahlung occurs when an electron penetrates the k shell and interact with the nucleus, the velocity of the electron is deflected and slowed down, the kinetic energy lost is emitted directly in the form of a photon radiation with a continuous spectrum with a maximum corresponding to the kVp of the HV; more than 80% of the X-rays emitted by a diagnostic tube are in the continuum with a maximum energy determined by the HV.

As the tube voltage increases, both the width and height of the spectrum broaden. The intensity, which is proportional to $kV^2 \cdot mA$, also rises. Additionally, the emission efficiency, defined as the ratio of X-ray output to the electrical power supplied, improves with increasing tube voltage (kV) and the atomic number (Z) of the anode.

Characteristic radiation occurs when an electron emitted by the filament interacts with the anode by expelling an electron on a specific orbital, for example the k-shell, and the subsequent rearrangement of other electrons on other orbitals, for example from the L or M shell. Due to this process a photon is emitted with a characteristic energy equal to the difference in the binding energies of the two shells (*i.e.*, $K_\alpha = E_K - E_L or K_\beta = E_K - E_M$). If the anode is made of tungsten(Z=74) $K_\alpha = 58keV and K_\beta = 68keV$, while if the material used is molybdenum(Z=42) $K_\alpha = 17.5keV K_\beta \sim 20keV$.

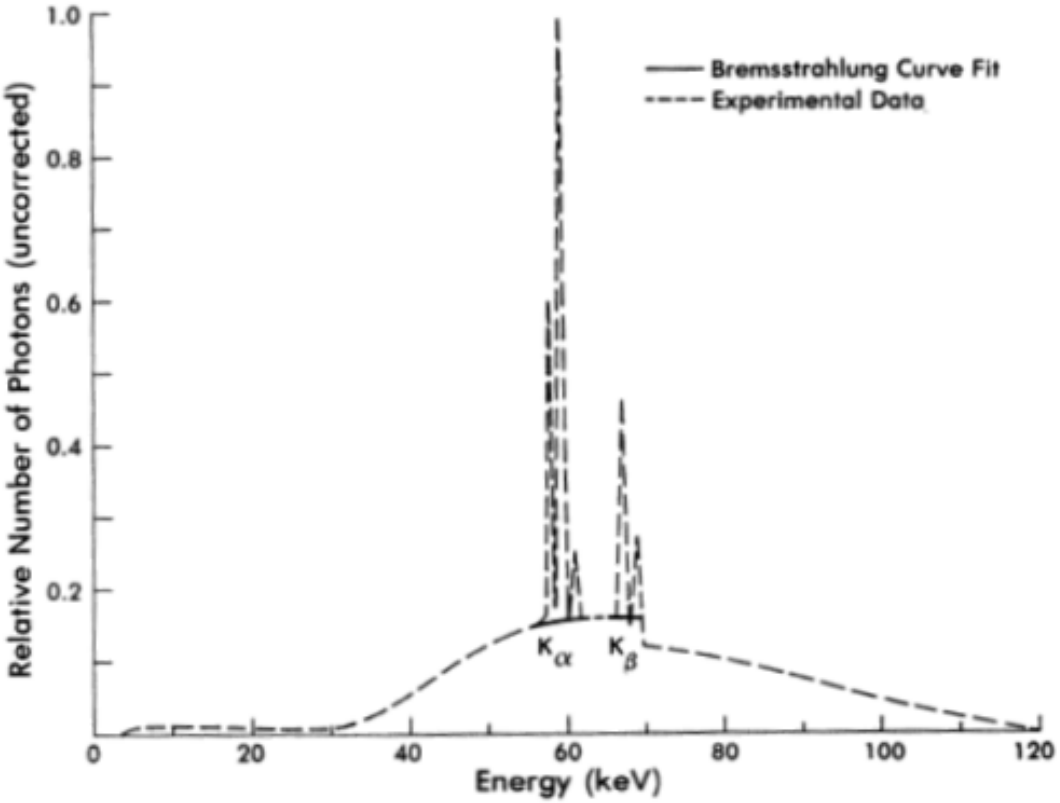


Figure 1.10: Separation of K X-rays from spectral data. The continuous Bremsstrahlung plot is obtained by fitting a curve to the data points that are not influenced by the K X-rays peaks.[40]

When the anode material has a high Z the Bremsstrahlung process is predominant, while if it has low Z the characteristic radiation assumes greater importance. For example in mammographic imaging Molybdenum anode tubes (Z=42) are more commonly used with low kVp and the characteristic X emissions are predominant with energies of 17.5 keV and 19.6 keV.

The lower energy photons in an X-ray beam are primarily absorbed by the patient's tissues, where they deposit radiation dose without significantly contributing to image formation. To minimize this unnecessary dose, a large portion of these low-energy photons is removed using filters as can be seen in Fig. 1.11, thus optimizing the balance between image quality and patient safety. Filtration improves the quality of the X-ray beam by increasing the ratio of useful, higher energy photons to lower energy photons, which only increase patient dose without enhancing the image.

Filtration can be classified as inherent or additional. Inherent filtration occurs due to the materials that the X-rays must pass through before exiting the X-ray tube, such as the target material, the glass envelope, and the insulating oil. The amount of inherent filtration is typically measured in aluminum equivalents, which refers to the thickness of

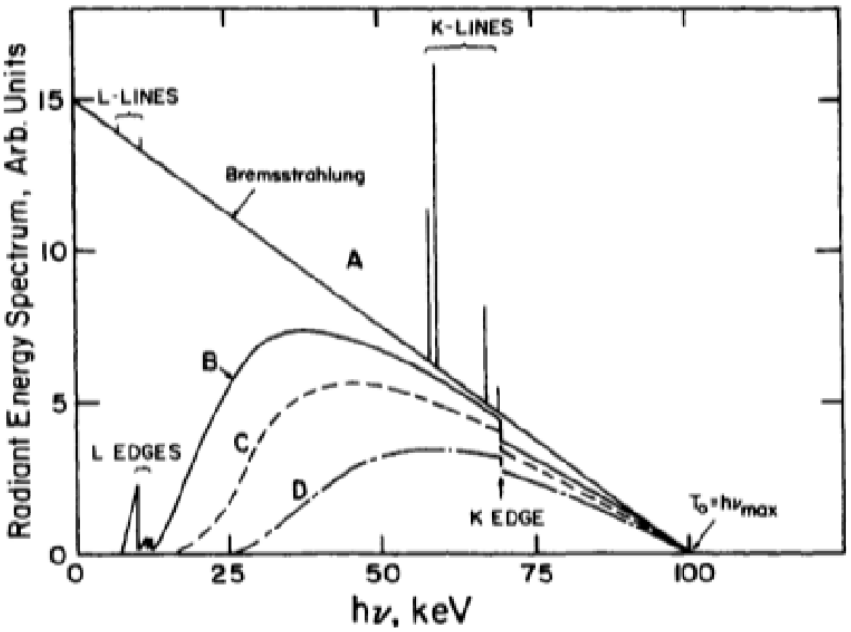


Figure 1.11: Typical X-ray spectra produced by 100 keV electrons, with increasing levels on filtration from A - D. A: Unfiltered. B: Filtered through 0.01 mm W in escaping the target. C: Additionally filtered through 2 mm Al. D: Filtered through 0.15 mm Cu and 3.9 mm A1 in addition to inherent target filtration. To avoid confusion, the K-fluorescence lines are not shown in curves B, C, and D, but are attenuated from their heights in curve A in the same proportion as the Bremsstrahlung is attenuated at the same energies.

aluminum that would produce the same attenuation. For most diagnostic X-ray systems, the inherent filtration is typically between 0.5 mm and 1 mm aluminum equivalent.

Another important factor is the heel effect, which arises due to the geometry of the X-ray tube and the target. As electrons penetrate the anode, X-rays emitted toward the anode side of the beam are more attenuated than those emitted toward the cathode side. This results in a beam intensity that is lower on the anode side, creating a non-uniform X-ray field. The effect is more pronounced with a smaller anode angle and results in a greater variation in intensity and an increase in the half-value layer (HVL)[1] toward the anode side.

Additional filtration is typically provided by uniform flat sheets of metal, commonly aluminum or copper, placed between the X-ray tube and the patient. These filters selectively

[1]In the context of narrow beam or "good geometry," the half-value layer (HVL) can be defined as the thickness of a specific material required to reduce the intensity of an X-ray beam to half its original value. For example, two successive HVLs will reduce the beam's intensity by a factor of four. The HVL is inversely proportional to the linear attenuation coefficient μ, which means that as the atomic number Z of the material increases, the HVL decreases due to higher attenuation.

absorb low-energy photons while allowing higher energy photons, which are more effective in producing the image, to pass through. The predominant attenuation mechanism at these energies is photoelectric absorption, which decreases rapidly as photon energy increases (varying approximately with the inverse cube of photon energy). To effectively remove low-energy photons while maintaining the high-energy part of the spectrum, the material of the filter must have an appropriate atomic number. Aluminum (Z=13) is commonly used because it balances the need for photoelectric absorption without overly reducing the intensity of higher energy photons. Copper (Z=29) is also used, sometimes in combination with aluminum, for more effective filtration in higher energy X-ray systems.

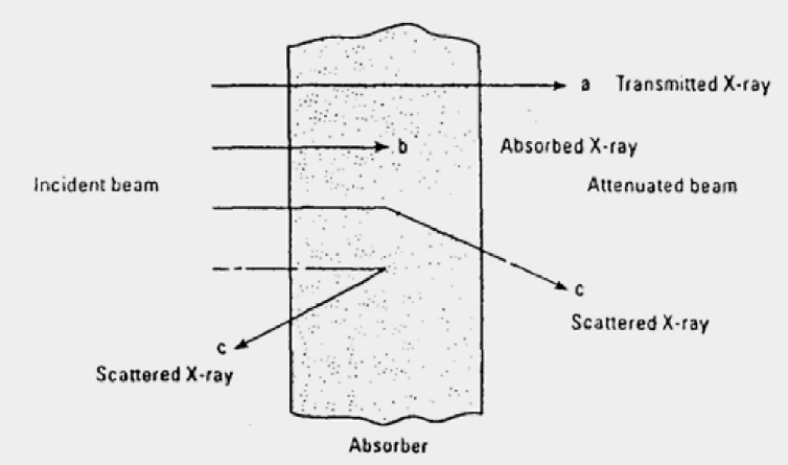


Figure 1.12: Interaction of X- or gamma rays with matter.

When high-energy X-ray photons pass through matter, as can be seen in Fig. 1.12, they can undergo three primary interactions: transmission, absorption, and scattering. In transmission, photons pass through the material without interacting with any atoms. This occurs when the photon's energy is not sufficient or the probability of interaction is low. The likelihood of transmission depends largely on the density and atomic composition of the tissue, as denser tissues present more opportunities for interactions.

In absorption, photons transfer all their energy to the atoms of the tissue, typically through the photoelectric effect. In this process, the photon is completely absorbed, ejecting an electron from the inner shell of an atom, and the photon ceases to exist. The probability of photoelectric absorption increases significantly with the atomic number (Z) of the material and is most likely to occur with lower-energy X-ray photons.

Scattering occurs when a photon's direction is altered by interaction with an atom, either with or without energy loss. In Compton scattering, the photon transfers part of its energy to an outer electron, causing the photon to change direction and continue with reduced energy. This is the dominant interaction for X-ray photons in the diagnostic energy range (20-150 keV). Rayleigh scattering, on the other hand, is an elastic scattering process where the photon changes direction without any loss of energy, although it is less significant for X-ray imaging.

Pair production, a high-energy interaction where a photon creates an electron-positron pair, only occurs at photon energies above 1.022 MeV, which far exceeds the energy range used in diagnostic radiology. Therefore, pair production does not occur in conventional X-ray imaging [41].

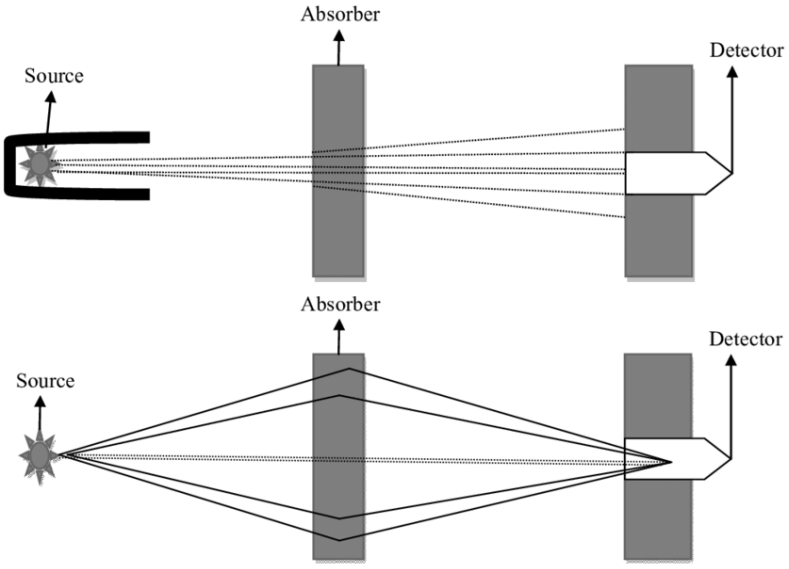


Figure 1.13: Attenuation of photon beam under conditions of narrow beam geometry (good geometry) and photon attenuation under conditions of broad beam geometry (bad geometry) [42]

The transmitted beam at any thickness of the absorber can be approximated by the exponential equation 1.1 only when in "good geometrical condition" as shown in Fig. 1.13,

$$I = I_0 e^{-\mu x} \tag{1.1}$$

where μ is the linear attenuation[2] coefficient of the absorber, I_0 is the initial intensity of incident photons and I is the transmitted intensity of the radiation and x is the thickness of the absorber. The probability of interaction depends on the photon energy, the material composition and its density. These parameters are incorporated in the mass attenuation coefficient that is measured in $\frac{cm^2}{g}$; that for $E\gamma \ll m_e c^2$:

$$\frac{\mu_{p.e.}}{\rho} \propto \left(\frac{Z}{h\nu}\right)^3 \;\&\; \frac{\mu_C}{\rho} \propto \frac{Z}{A} N_A \sim \sigma N_A/2 \tag{1.2}$$

For a wide X-ray beam, the percentage of photons transmitted through an object at a given distance is greater compared to a narrow beam. This is due to the contribution

[2]The linear attenuation coefficient measures the probability that a photon interacts (is absorbed or scattered) per unit length of the path it travels in a specific material, is a sum of the attenuation coefficients for each type of interaction $\mu = \mu_{p.e.} + \mu_C$

of scattered radiation, which increases the measured intensity. To account for this, the buildup factor BB can be defined as shown in Eq. 1.3:

$$B = \frac{I_{badgeometry}}{I_{goodgeometry}} \tag{1.3}$$

where $I_{badgeometry}$ refers to the intensity measured in the presence of scattered radiation (wide beam), and $I_{goodgeometry}$ represents the intensity measured in the absence of significant scatter (narrow beam) [43].

While the concept of HVL is primarily based on a monochromatic beam, it can also be applied to a polychromatic (non-monoenergetic) X-ray beam, where the exponential attenuation law no longer strictly applies. For a polychromatic beam, the HVL is defined as the thickness of material that reduces the initial beam intensity by a factor of one-half. As the beam passes through the material, beam hardening occurs due to preferential absorption of lower-energy photons. This process increases the proportion of higher-energy photons in the beam, making it more homogeneous and harder (more penetrating). As a result, the HVL is inversely proportional to the linear attenuation coefficient μ which in turn decreases with increasing photon energy.

The HVL is used to define the effective energy of the X-ray beam, which is the energy of a hypothetical monochromatic beam that would have the same HVL as the polychromatic beam. The effective energy is typically between one-third to one-half of the peak kilovoltage (kVp) of the beam.

1.2.3 Radiographic image

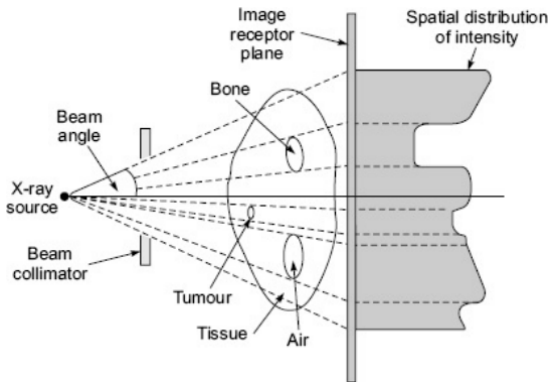
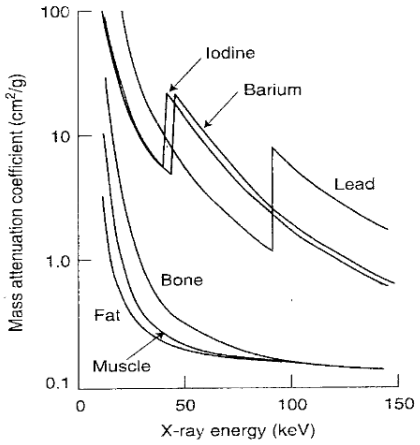


Figure 1.14: X-ray photons generated by the tube are directed at the patient. A fraction of the photons will reach the image receptor plane crating a 2-dimensional projection of the exposed anatomy. Due to different absorption coefficients of the materials.

The radiographic image is produced by the interaction of X-rays with a photon detector as shown in Fig. 1.14. The detector captures both primary photons, which pass through the patient without interaction, and secondary photons, which are scattered within the patient. However, only the primary photons contribute meaningful information to the image, as scattered photons degrade image quality by introducing noise.

$$\tau \propto \rho Z^3/E^3.$$



Fat	Air	Water, muscle	Bone
6.4	7.6	7.4	13.3

Figure 1.15: μ_m in respect to the X-ray energy. The plot shows soft tissues, bones, high Z materials (used like contrast agents), lead (used as a shield). The table below reports the effective Z of biological materials.

The probability of a photon passing through the patient without interacting is determined by the cumulative attenuation properties of the tissues on the photon path. Common attenuation coefficients as a function of the photon energy are shown in Fig. 1.15. Tissues with higher atomic numbers (Z) and densities, such as bone, have a greater capacity to absorb X-rays due to the higher probability of photoelectric absorption and Compton scattering. Thus, bones attenuates more photons, appearing bright (white) on the radiographic image.

In contrast, soft tissues like the liver, fat, and muscles are less dense and have lower atomic numbers. As a result, photons have a higher probability of passing through these tissues with little or no interaction. This leads to a greater number of transmitted photons reaching the detector, causing areas corresponding to soft tissue or air to appear darker on the radiographic image. The overall result is a 2D shadow projection where regions with more photon transmission, such as those representing soft tissues or air-filled spaces, appear darker, while areas with high photon absorption, such as bone, appear lighter [44].

Primary photons carry useful information and their distribution on the detector represents a measure of the attenuating properties of the tissues. Secondary photons, on the other hand, are deflected from their original path and do not carry useful diagnostic information, leading to a degradation of image quality. Several factors influence the quality of a radiographic images: unsharpness, contrast, noise and distortion and artifacts [45].

Unsharpness U_T , or the blurring of image details, is caused by various factors, the first one is the geometric unsharpness U_g influenced by the size of the radiation source, smaller the focal spot size (f) smaller U_g, the distances between object (patient) and image

receptor,smaller object-to-receptor distance (h) smaller U_g, and the distances between source and patient, smaller this distance (F-h) bigger U_g; so $U_g = fh/(F-h)$; when magnification is not needed, the test specimen is usually placed as close as possible to the detector and the source is placed some distance from the sample to minimize the penumbra; a greater distance between the source and the object will reduce geometric unsharpness. However, the intensity of the source decreases at the second power as the distance increases. Therefore, the source should be placed only as far away as necessary to control the penumbra.

The second one is the subject unshurpness U_s, some structure can be distinguished anatomically from its surroundings only by characteristics that vary gradually over distance, also, the shape of an object may prevent the projection of sharp boundaries onto the image receptor U_s is the result of the composition of the object, its shape, or a combination of both.

Motion unsharpness U_m is often a major contributor to unsharpness in a radiologic image. Motion causes boundaries in the patient to be projected onto different regions of the image receptor while the image is being formed; as a result, the boundaries are spread over a finite distance, and the resulting borders are blurred in the image. Voluntary motion often can be controlled by keeping examination times short and asking the patient to remain still during the examination. Motion can be "stopped" in the image by the use of very short examination times. In chest images, for example, examination times of a few milliseconds are used to gain a reasonable picture of the cardiac silhouette without the perturbing influence of heart motion; like also in studies of gastrointestinal tract.

Last one is receptor unsharpness U_r, in every display technique the image receptor inevitably adds unsharpness to the image; principally by the thickness and composition of the light-sensitive emulsion of the intensifying screens. These characteristics influence not only receptor unsharpness but also the sensitivity of the screens to X-rays, with increasing thickness, the sensitivity improves and the unsharpness increases. The choice of screens is, consequently, a trade-off between unsharpness introduced by the receptor and that resulting from motion caused by the finite time to record the imaging data. Summing all $U_T = \sqrt{U_g^2 + U_s^2 + U_m^2 + U_r^2}$ [46].

Contrast is defined as the ability to distinguish between different regions of the image, is defined in terms of the relative intensity change produced by an object. Is divided into various factors, one is the intrinsic contrast, therefore, structures in the patient can be distinguished in an image because they differ in physical composition. Some structures (e.g., breast) exhibit very subtle differences in composition and are said to have low intrinsic contrast; other structures (e.g., chest) provide large differences in physical density and atomic composition and yield high intrinsic contrast. Mathematically the patient is represented as in Fig. 1.16 by a uniform block of tissue of thickness t and linear attenuation coefficient μ_1, containing an embedded block of 'target' tissue of thickness x and linear attenuation coefficient μ_2. When a patient is exposed to a fairly uniform beam of X-radiation. The intensity of the transmitted photons depends on the different material through which they pass. Therefore, following the example shown in Fig. 1.16 there will

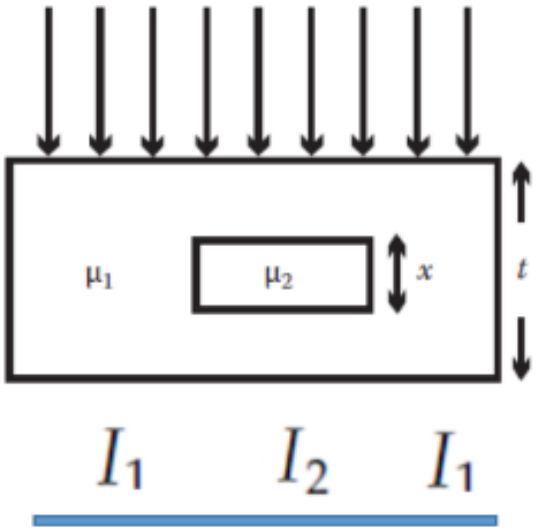


Figure 1.16: Simple model of the patient

be two intensities:

- $I_1 = N\epsilon E \exp -\mu_1 t + \epsilon_s \bar{E}_s S$
- $I_2 = N\epsilon E \exp -\mu_1 (t - x) - \mu_2 x + \epsilon_s \bar{E}_s S$

Consequently the contrast C of the 'target' tissue is expressed as in Eq. 1.4.

$$C = \frac{I_1 - I_2}{I_1} = \frac{1 - \exp{(\mu_1 - \mu_2)x}}{1 + R} = C_p D_s \tag{1.4}$$

Where R is the scatter-to-primary ratio, C_p the contrast given by the primary photons and D_s the degradation factor due to scattering. From Fig. 1.15 it can be noted that μ does not change significantly when the photon energy is high. Thus, contrast decreases rapidly with increasing photon energy. Consequently, to obtain a good contrast we should use a low photon energy. However, lower photon energies deposit a higher dose to the patient (which should always be minimised). Therefore, a compromise must be reached between image quality and radiation protection of the patients.

Contrast can also be increased by changing the material composition of the patient's inner structure. This is achieved by using contrast agents, which are substances that can be introduced into the body to better distinguish various tissues by providing a different signal. For example, in angiography, a water-soluble agent containing iodine (Z= 53, Ek =33 keV) is injected into the circulatory system to enhance the contrast of blood vessels by increasing the attenuation of X-rays impinging on the vessels.

Contrast influences spatial resolution (i.e. size of the smallest visible detail) defined as the ability to see fine detail. Which can clearly be seen when the contrast between the feature and its background is high. Resolution is quantified as the highest occurring frequency of lines that can be resolved in a high-contrast bar pattern.

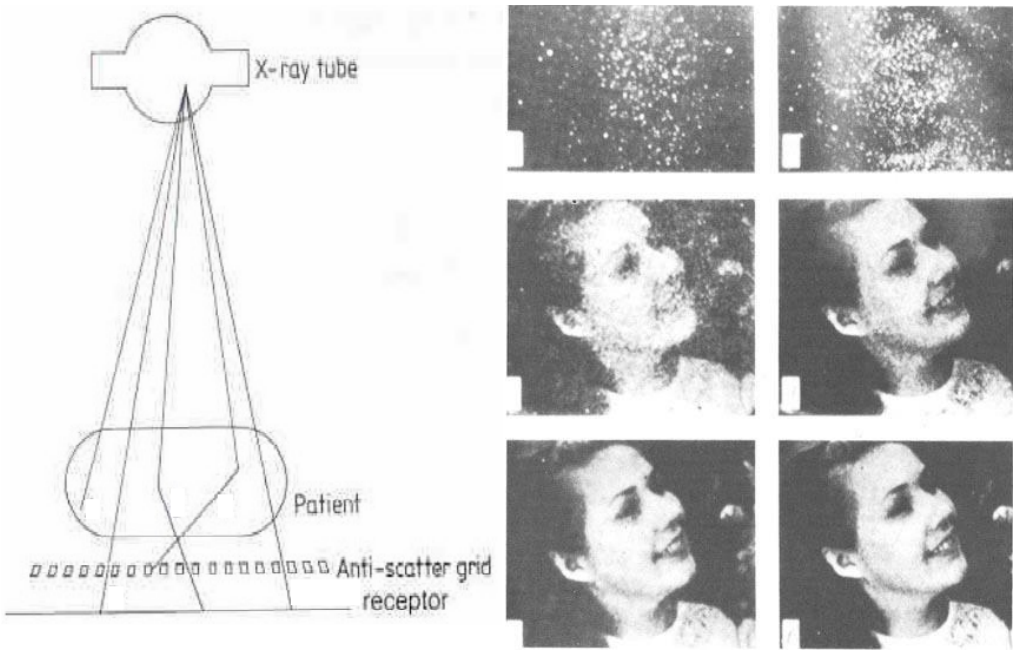


Figure 1.17: On the left there a picture of how the anti-scatter grid works, while on the right a depiction of the "quantum mottle" effect. As the illumination of the object increases, quantum mottle decreases [47].

Radiographic images are affected by noise, which is given by random photons scattered while traveling through an object. These photons will contribute to the background of the image without giving any relevant. The only effect being the degradation of the contrast. Radiographic noise has four possible components:

- **Structure noise:** information about the structure of the patient that is unimportant to diagnosis and characterization of the patient's condition, for example, shadows of the ribs can hide small lesions under them.
- **Receptor noise:** receptors are not uniformly sensitive to radiation over their active surfaces, they impose a pattern of receptor noise onto the image; it is rebalanced via computer after the acquisition.
- **Quantum mottle:** is given from the finite number of information carriers used to create the image. Generally this is the dominating type of noise, and goes like $\sqrt{N}$ from statistical laws. Where N is defined as the number of photons that generate the signal.
- **Radiation noise:** information present in a radiation beam that does not contribute to the usefulness of the image; for example nonuniform intensity of the X-ray beam given by the Heel effect, or more important the scattered radiation.

The noise contribution to the primary photon intensity on the detector is given by the scatter to primary ratio R shown in Eq. 1.4. This parameter, is in first approximation

proportional to the thickness of the object, the field size and the energy of the photons. By placing an anti-scatter grid between the patient and the image receptor, it is possible to reduce the contribution of non primary photons on the detector as schematized in Fig. 1.17. This is achieved by absorbing scattered photons and by not deflecting the primary beam. A common implementation of anti-scatter grid is made by 0.05 mm thick lead strips.

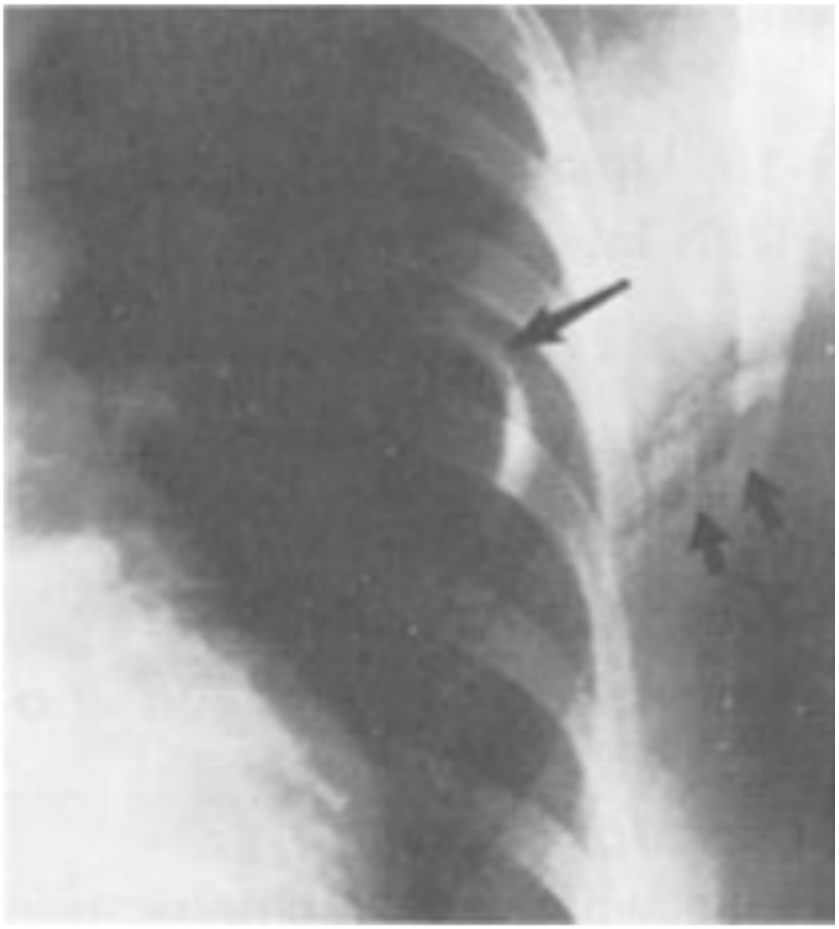


Figure 1.18: Examples of image radiographic artifacts caused by crinkling the film (single arrow) and exerting pressure on the film (double arrows) [48] [49]

Distortion and artifacts are present to some degree in every diagnostic image. Image distortion is caused by unequal magnification of various structures in the image. Artifacts can arise from so many causes that their complete description is not possible here.An example are streaks caused by moving structure as shown in Fig. 1.18.

1.2.4 From Radiographic Imaging to Computed Tomography

Conventional radiographic images are fundamentally projections that sum the attenuation coefficients of all tissues along the path of the X-ray beam. This leads to a loss of depth

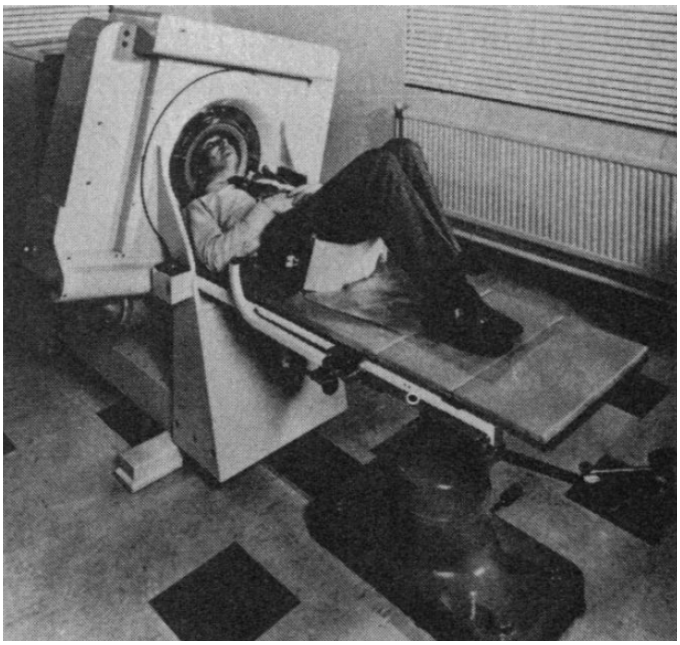


Figure 1.19: First clinical prototype brain scanner installed at Atkinson Morley's Hospital, London.[50]

information, as all structures in the X-ray path are superimposed onto a single plane (i.e. it merely shows the integrated attenuation profile). Furthermore, there is a significant limitation in the ability to differentiate between tissues with similar X-ray attenuation coefficients. The contrast achievable with traditional X-ray films is about 2%, which is sufficient for differentiating high-contrast structures like bone (which shows a -26% attenuation difference compared to muscle) or air ($+20\%$), but not for soft tissues like blood and muscle where differences are less than 1%. To overcome these problems, computed tomography (CT), emerged as a significant advancements by providing three-dimensional insights into the anatomy, allowing for better discrimination of structures with similar attenuation coefficients and more accurate localization of anatomical features.

The fundamental principle of CT is based on acquiring multiple X-ray projections from different angles around the patient and reconstructing these data into cross-sectional images. These reconstructed images provide detailed anatomical information of thin slices through the body, eliminating the problem of superimposition of structures and significantly improving contrast resolution.

The initial CT scanners as one seen in Fig.1.19, utilized a pencil beam and a single detector to scan the patient in a translational and rotational manner. The scanning process was time-consuming, and the image quality was limited by the simple detector technology and computational capacity of the time. The introduction of second-generation CT scanners, which employed a narrow fan beam and multiple detectors, reduced scan times and improved image quality by allowing multiple projections to be acquired simultaneously [52].

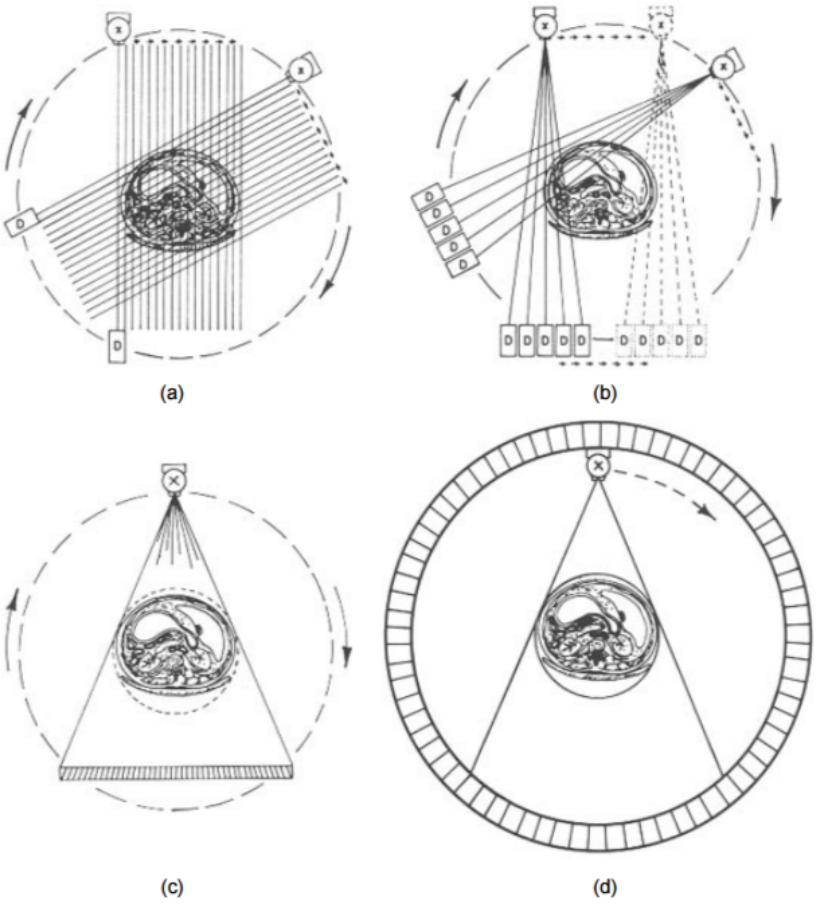


Figure 1.20: Scan motions in computed tomography. A: first-generation scanner using a pencil x-ray beam and a combination of translational and rotational motion. B: Second-generation scanner with a fan x-ray beam, multiple detectors, and a combination of translational and rotational motion. C: Third-generation scanner using a fan x-ray beam and smooth rotational motion of x-ray tube and detector array. D: Fourth-generation scanner with rotational motion of the x-ray tube within a stationary circular array of 600 or more detectors.[51]

Third-generation CT scanners employ a wide fan-shaped X-ray beam that covers the entire width of the patient and a curved array of detectors that rotate synchronously with the X-ray tube around the patient. This design allows for continuous data acquisition as the gantry rotates smoothly through a full 360 degrees. Unlike previous generations, where data were acquired in a stop-and-start way, third-generation scanners use a slip-ring technology that enables continuous rotation of the gantry without the need for cables, significantly reducing scan times. The continuous data acquisition allows for more advanced image reconstruction algorithms, such as iterative reconstruction techniques, which further improve image quality by reducing noise and enhancing contrast resolution. Third-generation CT scanners also introduced the concept of the helical (or spiral)

scan, where the patient is moved continuously through the rotating gantry. This allows for the acquisition of volumetric data in a single breath-hold, reducing respiratory motion artifacts and providing high-quality images of large anatomical regions. Helical CT has become the standard for many clinical applications, including lung cancer screening, trauma imaging, and cardiac CT angiography, where rapid and high-resolution imaging is essential.

The evolution seen in Fig.1.20 of detector technologies has been a key factor in advancing CT capabilities. Third-generation CT scanners initially used scintillation detectors coupled with photomultiplier tubes, but modern scanners now employ solid-state detectors, which offer several benefits. Solid-state detectors are more efficient in converting X-ray photons to electrical signals, which improves image quality by reducing noise and increasing contrast resolution. They also have a faster response time, which allows for quicker data acquisition and shorter scan times[53].

Moreover, the development of multi-slice CT (MSCT) in the late 1990s marked another significant advancement in CT technology. MSCT scanners use a multi-row detector array that enables the simultaneous acquisition of multiple slices in a single rotation. This innovation drastically reduces the time required for complete anatomical coverage and allows for the acquisition of isotropic voxels, which provide high-resolution images in all three planes (axial, coronal, and sagittal). Modern MSCT scanners can acquire up to 320 slices per rotation, making them capable of imaging entire organs like the heart or brain in just a few seconds.

The latest developments in detector technology include the introduction of dual-source CT scanners, which use two X-ray tubes and two detector arrays mounted on the gantry at 90 degrees to each other. This configuration allows for faster data acquisition, reduced motion artifacts, and the ability to perform dual-energy imaging to obtain additional information about tissue composition.

While CT has become the gold standard for many diagnostic applications, other advanced imaging techniques such as Digital Tomosynthesis (DTS) and Cone Beam Computed Tomography (CBCT) offer specialized advantages in certain clinical contexts.

Digital Tomosynthesis (DTS) as can be seen in Fig. 1.21 is a hybrid imaging technique that combines elements of both conventional radiography and CT. DTS involves acquiring multiple low-dose X-ray projections over a limited angular range, which are then reconstructed into a series of slices using a computational algorithm to provide high-resolution images of specific planes or slices, making it particularly useful in breast imaging (mammography).

Cone Beam Computed Tomography (CBCT), on the other hand, has become increasingly popular in dental and maxillofacial imaging due to its ability to provide high-resolution images at a lower radiation dose compared to conventional CT. Unlike traditional CT that uses a fan beam, CBCT utilizes a cone-shaped X-ray beam and a flat-panel detector to acquire volumetric data in a single rotation around the patient. This allows for the creation of detailed 3D reconstructions of the area of interest, such as teeth, or sinuses.

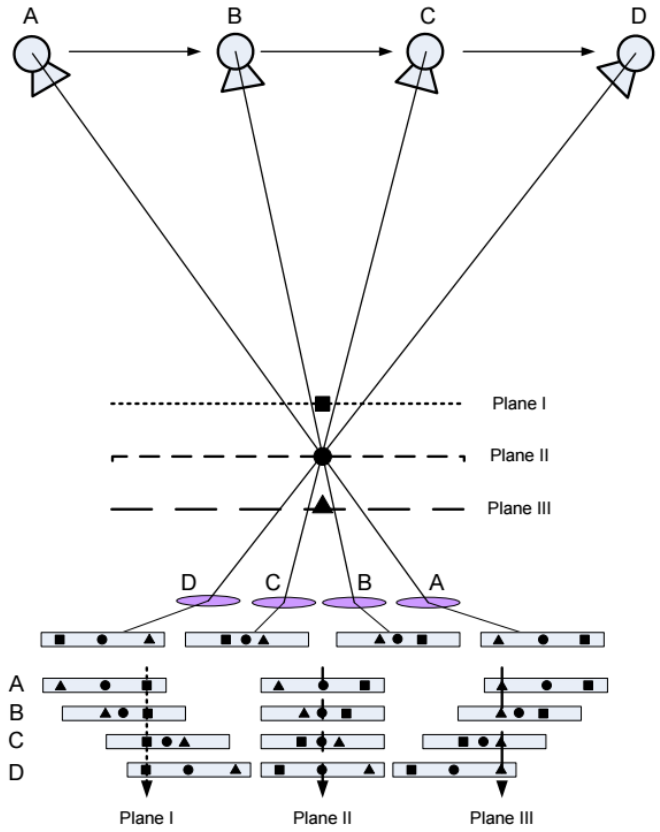


Figure 1.21: The X-ray tube moves at a constant speed to the right, while the image receptor moves at a constant speed to the left. In this figure, four samplings of the image are shown at tube positions A, B, C and D. By combining the four sampled images with appropriate offsets, it is possible to create tomographic images focused on planes I, II and III as indicated.[54]

The NewTom 7G, for example, is a state-of-the-art CBCT system that offers extensive clinical applications across multiple specialties, including orthopedics, otolaryngology, and maxillofacial surgery. It provides ultra-high-definition images with voxel resolutions as fine as 90 microns, making it ideal for examining small bone structures, assessing joint mobility, and performing post-operative evaluations with minimal artifacts. The system's ability to adjust the field of view (FOV) and X-ray dose based on patient size and diagnostic needs further enhances its utility, allowing for tailored imaging protocols that minimize radiation exposure [55].

Digital CT imaging allows for the use of Hounsfield Units (HU) to quantify the attenuation properties of different tissues. The Hounsfield Unit is defined as Eq.1.5:

$$\text{CT-Number} = \left(\frac{\mu_{\text{tissue}} - \mu_{\text{water}}}{\mu_{\text{water}}} \right) \times 1000 \tag{1.5}$$

where μ_{tissue} is the linear attenuation coefficient of the tissue and μ_{water} is the linear

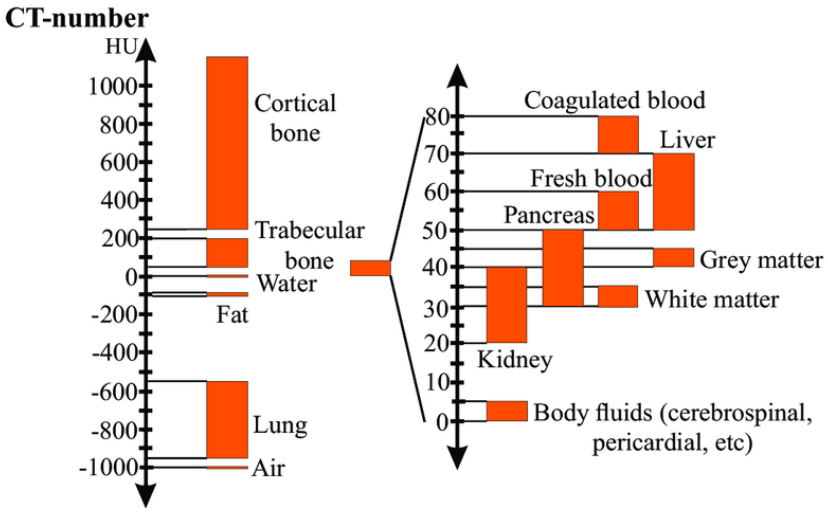


Figure 1.22: Hounsfield-Scale for different kinds of tissues.[56]

attenuation coefficient of water. This unit provides a standardized way to differentiate between various tissues based on their density and composition as can be seen in Fig. 1.22. Windowing techniques in digital imaging involve selecting a specific range of Hounsfield Units to display as a grayscale image. By adjusting the window level and width, radiologists can focus on different tissue types or abnormalities, enhancing the diagnostic utility of the image. The radiodensity, measured in Hounsfield Units (HU, also known as CT number) is inaccurate in CBCT scans because different areas in the scan appear with different greyscale values depending on their relative positions in the organ being scanned[57]; HU-based differentiation of material do not apply to cone beam computed tomography (CBCT) scans, as CBCT scans provide unreliable HU readings [58].

1.2.5 Image Reconstruction and Processing

The reconstruction of images in CT is a computational process that converts raw data from X-ray attenuation measurements into cross-sectional images [59].

To reconstruct the CT Image from the acquisition of a complete set of projections at various angles, the software has to use the basic principle that X-ray attenuation can be represented as a line integral through the object. Mathematically, the attenuation I_j measured by the detector element j, using the Eq. 1.1, can be expressed as:

$$I_j = I_0 \exp\left(-\int \mu(x,y)ds\right) \tag{1.6}$$

where I_0 is the initial intensity of the X-ray beam,$\mu(x,y)$ is the linear attenuation coefficient at position (x,y), and the integral is taken along the path of the X-ray through the object.

Before image reconstruction, several preprocessing procedures are applied to the raw projection data to correct for various imperfections and noise. These steps may include:

- **Normalization**: During routine calibration of the CT scanner, air scans are performed to characterize the influence of the individual detector responses. The measured projection data for a given scan are normalized by these calibration scans to correct for previously identified inhomogeneities in the field. This procedure can be mathematically represented as:$P_j = \frac{I_j}{I_r}$ where I_r is the signal measured by a reference detector located outside the field of view (FOV) of the patient.
- **Dead Pixel Correction**: In some scanners, a fraction of the detector elements may be "dead" (non-responsive). These are routinely identified, and a dead pixel correction algorithm is applied, replacing dead pixel data with interpolated data from surrounding pixels.
- **Scatter Correction**: Scatter correction algorithms are generally applied before the logarithmic transformation of the data. These algorithms aim to reduce the impact of scatter radiation, which can cause artifacts and degrade image quality.
- **Adaptive Noise Filtration**: Methods such as adaptive noise filtration identify regions in the projection data that correspond to low signal areas, which are typically associated with high noise. Smoothing or other data processing steps are applied to these regions to reduce noise in the final reconstructed image.

Following preprocessing, the projection data undergo logarithmic transformation and normalization accounting for the characteristics exponential attenuation of X-ray interactions. This process linearizes the relationship between the measured intensity and the attenuation coefficients, simplifying the reconstruction process. The transformed projection data, P_j, can be expressed as:

$$P_j = \int \mu(x, y) ds \tag{1.7}$$

and represents the sum of the attenuation coefficients along the path of the ray through the object. In CT imaging, as shown in Fig. 1.23 two coplanar reference frames are defined:

- The XY frame, which is fixed with respect to the object.
- The X_rY_r frame, where the Y_r direction is aligned with the incident X-ray beam.

The origin of both systems is located at the center of rotation of the scanning gantry. The X_rY_r frame is rotated by an angle θ counterclockwise with respect to the XY frame. In the polar representation, also known as Radon-space, each point represents a line integral taken through the object. The transformation between object space and projection space can be expressed using the Radon transform:

$$R(\rho, \theta) = \int_{-\infty}^{\infty} f(x \cos\theta + y \sin\theta) ds \tag{1.8}$$

R represents the projection data in Radon space.

The simplest reconstruction algorithm is the backprojection method (BP) which projects the measured data back across the image plane. However, this simple approach results in a blurred image. To correct for this blurring, a mathematical filtering operation is required,

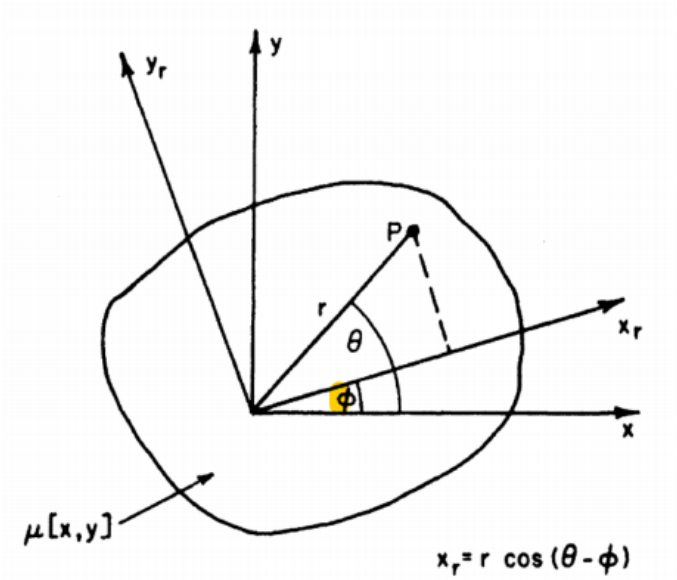


Figure 1.23: Object is presented as a two-dimensional distribution of linear attenuation coefficient $\mu[x, y]$. The X-ray source and detector rotate with the x_r-y_r frame, with the X-rays traveling parallel to y_r. P is the general point of the object.

known as filtered backprojection (FBP). The FBP, that can be seen in Fig. 1.24, which involves the following steps:

1. **Filtering**: The raw projection data are convolved with a filter function to compensate for the blurring effect of simple backprojection. The convolution operation in one dimension is defined as:

$$(f * h)(x) = \int_{-\infty}^{\infty} f(t)\, h(x - t)\, dt \tag{1.9}$$

   where $h(x)$ is the convolution kernel.

2. **Backprojection**: The filtered projections are then backprojected onto the image plane to reconstruct the image. The backprojection operation for each filtered projection $g_\theta(x_r)$ is given by:

$$f(x, y) = \int_0^{\pi} g_\theta(x \cos\theta + y \sin\theta)\, d\theta \tag{1.10}$$

Another reconstruction techniques, such as model-based iterative reconstruction (MBIR), provide a more sophisticated approach to image reconstruction. These techniques are computationally intensive and refine the image quality by iteratively comparing the reconstructed image to the raw data and adjusting the image to minimize differences. The iterative process can be described as follows:

1. **Initial Estimate**: The reconstruction process begins with an initial estimate of the image, which may be a constant image or an image reconstructed using FBP.

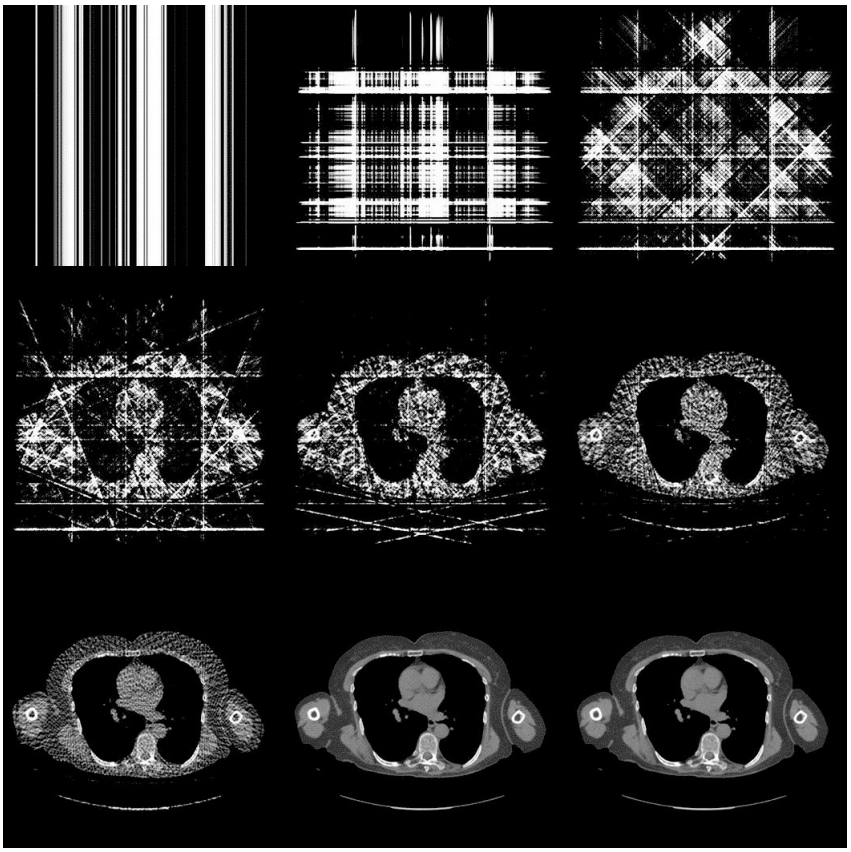


Figure 1.24: Filtered back projections can be used to achieve a good reconstruction of the space domain. The images are associated with, respectively, 1, 2, 4, 8, 16, 32, 64, 256 and 1024 filtered back projections at different angles.[60]

2. **Forward Projection**: From this initial estimate, synthetic projection data are generated using forward projection. This step simulates the actual imaging process and produces projection data from the estimated image.

3. **Error Calculation**: The synthetic projection data are compared with the measured projection data to calculate an error matrix for each projection angle.

4. **Image Update**: The image estimate is updated based on the error matrix to reduce the differences between the synthetic and measured projection data. The update step can be mathematically represented as:

$$f^{(k+1)}(x,y) = f^{(k)}(x,y) + \lambda \sum_{\theta} \left(P_{\text{measured},\theta} - P_{\text{synthetic},\theta}\right) W(x,y,\theta) \tag{1.11}$$

where $f^{(k+1)}(x,y)$ is the updated image estimate, $P_{\text{measured},\theta}$ and $P_{\text{synthetic},\theta}$ are the measured and synthetic projections, respectively, and $W(x,y,\theta)$ is a weighting function that accounts for various factors such as noise characteristics and system response.

5. **Convergence Check**: The process is repeated iteratively until the image converges to a solution that minimizes the error matrix, resulting in a reconstructed image with enhanced noise reduction and spatial resolution.

The filtered backprojection process can be efficiently implemented using the Fourier transform [61]. The convolution theorem states that convolution in the spatial domain is equivalent to multiplication in the frequency domain [62]. Using the Fourier transform approach allows for faster computation, as it avoids direct convolution in the spatial domain. This approach is particularly advantageous for commercial CT scanners, which require rapid processing capabilities.

In Cone-beam computed tomography the X-rays used are not collimated [63]. Therefore, the detector consists of a square surface, composed by scintillators and ccd pixel matrix. Compared to diagnostic CT systems, cone beam CT systems have a relatively slow rotation of several seconds to minutes per rotation. The most commonly used recording trajectory is a circular path. The implementation of a filtered backprojection, named after the authors Feldkamp, Davis and Kress as the FDK algorithm, is used as the standard reconstruction method; the disadvantage of which is that only the central layer fulfills the Tuy condition[3], meaning that only this layer contains mathematically complete data[64]

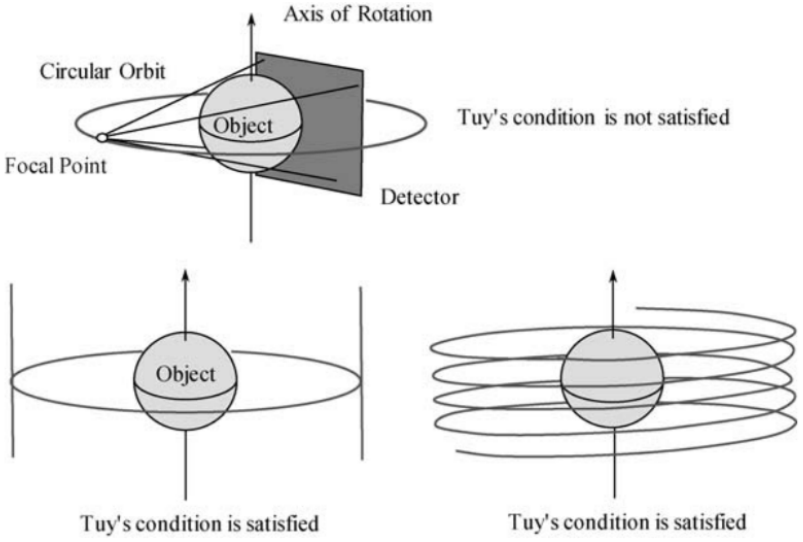


Figure 1.25: the circular cone-beam focal-point orbit does not satisfy Tuy's condition. If we draw a plane cutting through the object above (or below) the orbit plane and parallel to the orbit plane, this plane will never intersect the circular orbit. The helical and circle-and-lines orbits satisfy Tuy's condition, and they can be used to acquire cone-beam projections for exact image reconstruction[65].

Not undergoing to Tuy's sufficiency condition, causes artifacts in the reconstruction result

[3]Cone-beam data-sufficiency condition: Every plane that intersects the object of interest must contain a cone-beam focal point

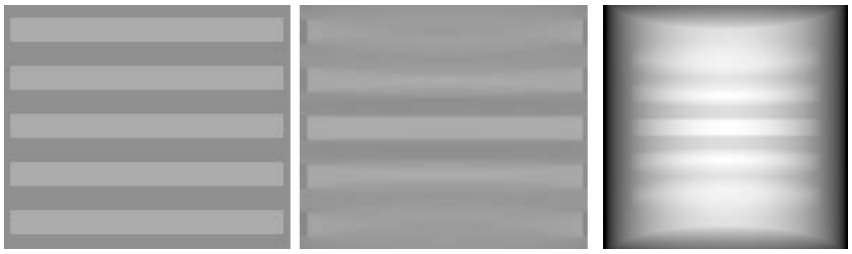


Figure 1.26: Effect of the cone-beam geometry in a circular trajectory with a cone angle of 30 degrees. From left to right: XZ slice of the original Defrise disk phantom, the projection image for all projection angles, and the XZ slice of the reconstructed volume using the FDK algorithm.[66]

that are called cone-beam artifacts as seen in Fig. 1.26. An elegant way to overcome this problem, is to rotate the source along a helical path, as a continuous motion is obtained as can be seen in Fig.1.25. Note that a change in the path of the trajectory increases the Dose to the patient and the time of the exam; also often implies a different reconstruction algorithm. implementing this is not the aim of the present work.

The choice of the reconstruction algorithm and processing technique depends on the specific clinical application and the desired balance between image quality and radiation dose. For example, in pediatric imaging or lung cancer screening, where minimizing radiation exposure is crucial, advanced iterative reconstruction (IR) or artificial intelligence (AI) based methods are preferred due to their superior noise reduction capabilities. In contrast, in emergency trauma settings where speed is of the essence, FBP may still be the method of choice due to its rapid processing capabilities.

OSCaR: Open Source Cone-beam Reconstructor

In this work the reconstruction of the CBCT image was obtained using "OSCaR: Open Source Cone-beam Reconstructor" [67]. OSCaR is an open-source Matlab FDK(Feldkamp-Davis-Kress) [64] tool whose development was supported by the American Association for Physicists in Medicine(AAPM) Imaging Research Subcommittee. It offers a Graphical User Interface (GUI) for CBCT reconstructions using series of 2D projections. There are three steps to use OSCaR: pre-processing, reconstruction and export.

To pre-process data, CBCT projections have to be MATLAB readable images(DICOM, JPG, or others); the projections must be stored in individual images and all the parameters should be specified by the user:

- N_{proj}: Number of projections
- N_{row}: Number of rows
- N_{col}: Number of columns
- **du**: Thickness of a pixel

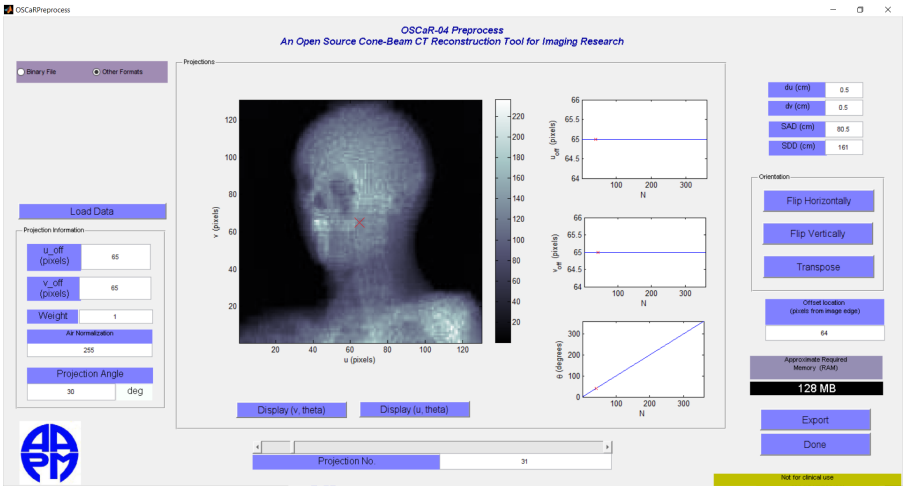


Figure 1.27: OSCaR Process GUI

- **dv**: Height of a pixel
- **SAD**: Source-Axis Distance
- **SDD**: Source-Detector Distance
- **.csv file**: Consists of N_{proj} rows, each with 6 columns:
  - **filename**: Name of the file corresponding to the kth projection
  - θ_G: Gantry angle of that projection
  - u_off: Offset of center of detector perpendicular to the axis in PIXELS
  - v_off: Offset of center of detector parallel to the axis in PIXELS
  - I_0: Air normalization
  - **w**: Weight of that projection

As can be seen in Fig. 1.27; after importing the projections and all the parameters the user has to export the correspondent MATLAB file.

Next, the reconstruction process can be initiated. The user needs to import the MATLAB file and define the borders of the region to be reconstructed. OSCaR will automatically determine the voxel dimensions and the number of slices. Additionally, the user can select a filter to use in the FBP algorithm before proceeding to reconstruct the DICOM CBCT image 1.28.

DICOM® - Digital Imaging and Communications in Medicine - is the international standard for medical images and related information. It defines the formats for medical images that can be exchanged with the data and quality necessary for clinical use. DICOM® is implemented in almost every radiology, cardiology imaging, and radiotherapy device (X-ray, CT, MRI, ultrasound, etc.), and increasingly in devices in other medical domains

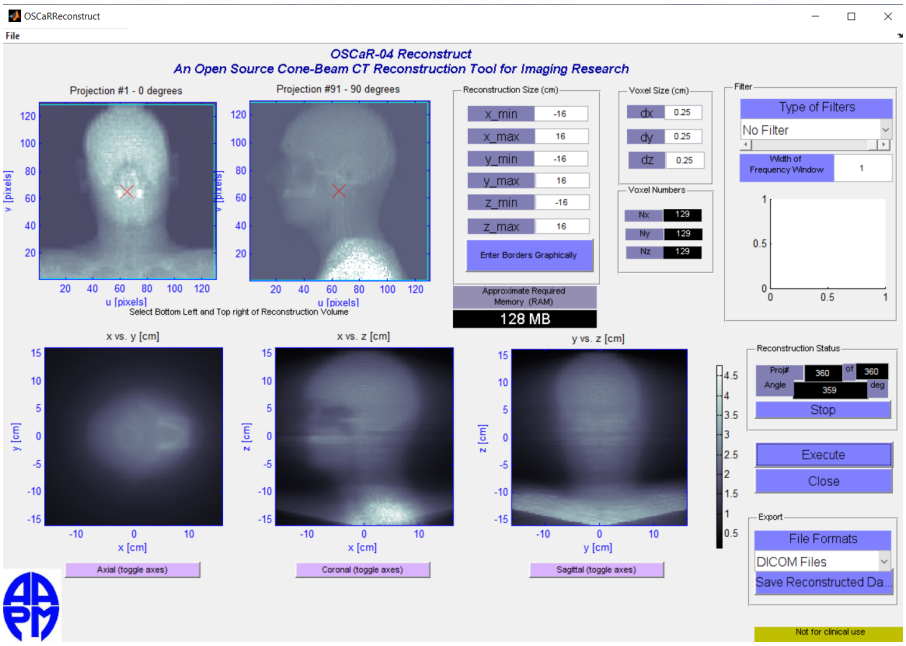


Figure 1.28: OSCaR Reconstruction GUI

such as ophthalmology and dentistry. With hundreds of thousands of medical imaging devices in use, DICOM® is one of the most widely deployed healthcare messaging Standards in the world. There are literally billions of DICOM® images currently in use for clinical care [68].

Chapter 2

Monte Carlo methods

In this chapter the Monte Carlo (MC) methods used to simulate the CBCT scan are discussed. First a historical and mathematical introduction to the computational method will be shown and then the application of Monte Carlo codes for these thesis work. In particular two codes were used for this research: GATEv.9.3 and Geant4-11.2.2. Both will be explained in detail taking into account their characteristics.

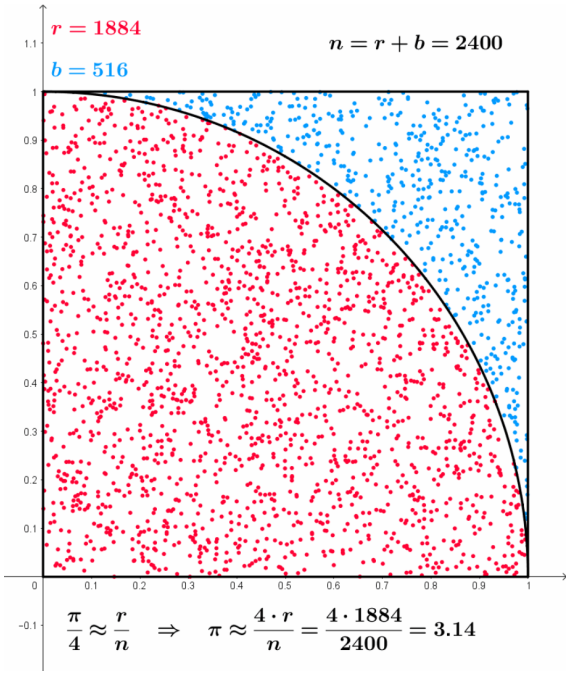


Figure 2.1: MC method applied to approximating the value of π[69].

Monte Carlo methods, refers to a class of computational algorithms that rely on random sampling to obtain numerical results. The essence of these methods is to use random variables to simulate complex phenomena and estimate quantities that are otherwise difficult

to calculate using traditional analytical techniques [70]. MC simulations are employed in a wide range of applications, from numerical integration to solving stochastic differential equations, and from optimization problems to statistical mechanics. These methods are particularly useful for tackling problems with a large number of variables or in scenarios where deterministic methods fail due to complexity or the need for approximations [71].

The power of MC methods lies in their flexibility to model random processes and their ability to approximate solutions by averaging over large numbers of random samples. As the number of simulations increases, the results obtained tend to converge to the true value, a property guaranteed by the Law of Large Numbers and the Central Limit Theorem.

2.1 Historical Overview

Figure 2.2: Stanislaw Ulam. This portrait is a work of Jeff Segler[72].

The theoretical foundations of MC methods can be traced back to 1777 when the French mathematician Georges-Louis Leclerc, Count of Buffon, conducted an experiment that involved throwing a needle onto a plane crossed by parallel lines. He used random samples to estimate the probability of the needle crossing a line, a process that laid the groundwork for modern MC methods. Buffon's work is recognized as the first recorded use of random numbers to solve a mathematical problem, specifically an integral [73].

More than a century later, the French mathematician Pierre-Simon Laplace extended

this idea by suggesting that the value of π could be estimated using a similar needle experiment, thus applying random sampling principles to determine a mathematical constant. However, due to the slow convergence of these early methods, practical applications remained limited until the advent of computers.

In the 1940s, John von Neumann and Stanislaw Ulam, the last one represented in figure 2.2, while working on the Manhattan Project, coined the term "Monte Carlo methods" in reference to the famous Monte Carlo Casino. They developed these methods to solve complex integrals involved in the study of neutron diffusion and the behavior of fissile materials. The introduction of computers allowed for the efficient execution of MC simulations, leading to their widespread adoption in fields ranging from nuclear physics to economics and finance [74].

After 1950, MC methods transitioned from being a mathematical curiosity to an indispensable tool in scientific research, largely thanks to advances in computing technology. Computers provided the means to quickly generate random numbers and execute long calculations that were previously impractical. Today, MC methods are used in diverse fields, including chemistry, physics, economics, and medicine.

2.2 Mathematical Foundations

MC methods are grounded in probability theory and rely on stochastic processes to simulate phenomena. These methods use random numbers to estimate quantities of interest, often involving the solution of integrals or sums over probability distributions.

Consider a random variable T associated with a stochastic process, represented as a function of k random variables $(X_1, X_2, \ldots, X_k)$:

$$T = f(X_1, X_2, \ldots, X_k).$$

The expected value of T can be expressed as the integral of T with respect to the probability density function (p.d.f.) of the random variables $(X_1, X_2, \ldots, X_k)$:

$$I = \int_D f(x_1, x_2, \ldots, x_k) p(x_1, x_2, \ldots, x_k) dx_1 \cdots dx_k,$$

where $p(x_1, x_2, \ldots, x_k)$ is the p.d.f. of the random variables. In practice, this integral is often difficult or impossible to evaluate analytically, particularly in high-dimensional spaces. MC methods approximate the value of the integral by generating random samples $(x_{1i}, x_{2i}, \ldots, x_{ki})$ from the distribution $p(x_1, x_2, \ldots, x_k)$ and computing the average of the function f over these samples:

$$T_N = \frac{1}{N} \sum_{i=1}^{N} f(x_{1i}, x_{2i}, \ldots, x_{ki}),$$

where N is the number of samples. The quantity T_N is an unbiased estimator of I, and as N increases, the estimate converges to the true value I. This convergence is guaranteed by the Law of Large Numbers, which states that the sample mean will converge to the expected value as the number of samples tends to infinity as can be seen in figure 2.1 [73].

2.2.1 The Law of Large Numbers

The Law of Large Numbers (LLN) is a fundamental theorem in probability theory that underpins the MC method. It states that as the number of independent random samples increases, the sample mean will converge to the true expected value of the distribution:

$$\lim_{N\to\infty} \overline{z} = \langle z \rangle,$$

where $\overline{z}$ is the sample mean and $\langle z \rangle$ is the expected value. The LLN provides the theoretical basis for the reliability of MC simulations, ensuring that, with a sufficiently large number of samples, the estimate will be close to the true value.

2.2.2 The Central Limit Theorem

The Central Limit Theorem (CLT) further strengthens the foundation of MC methods by describing the distribution of the sample mean for large N. Specifically, the CLT states that the distribution of the sample mean tends toward a normal distribution with mean μ and variance σ^2/N, where μ and σ^2 are the mean and variance of the original random variable:

$$P\left(|T_N - I| \leq \frac{3\sigma_T}{\sqrt{N}}\right) \approx 0.997.$$

This result implies that, as N increases, the probability that the MC estimate is close to the true value I approaches 1. However, the convergence rate of MC methods, typically proportional to $1/\sqrt{N}$, means that a large number of samples may be required to achieve a desired level of accuracy.

2.3 Advantages and Limitations

MC methods have several key advantages:

- **Simplicity**: Once the random variables are defined, the process of sampling and averaging is straightforward.
- **Versatility**: MC methods can be applied to a wide range of problems, from evaluating multidimensional integrals to optimizing complex systems.
- **Scalability**: These methods can handle problems with a large number of variables, where traditional deterministic methods may fail.

Despite these advantages, MC methods also have notable limitations:

- **Slow Convergence**: The rate of convergence is typically proportional to $1/\sqrt{N}$, meaning that a large number of samples is required to achieve high precision.
- **Computational Cost**: For complex systems, running a sufficient number of simulations can be computationally expensive.

Various variance reduction techniques, such as importance sampling and stratified sampling, can be employed to improve the efficiency of MC simulations. These methods aim

to reduce the variance of the estimator without increasing the number of samples, thereby improving the accuracy of the estimate.

2.4 Applications of MC Methods

MC methods are used in a wide array of fields, including:

- **Nuclear Physics**: Simulating particle interactions, neutron diffusion, and absorption in fissile materials, as well as radiation shielding.
- **Finance**: Pricing complex financial derivatives and assessing risk in uncertain markets.
- **Medicine**: Optimizing radiation treatment plans, simulating biological processes, and modeling the spread of diseases.
- **Economics**: Analyzing economic models under uncertainty and simulating market behavior.
- **Chemistry**: Modeling molecular dynamics and chemical reactions in complex systems.

MC methods have become indispensable tools in many scientific and engineering disciplines, where they are often the only feasible solution for solving high-dimensional, nonlinear, or stochastic problems.

2.5 GEANT4 & GATE Monte Carlo Codes

MC codes used are Geant4-11.2.2 and GATEv9.3.

GEANT4 (GEometry ANd Tracking 4) [76] [77] [78] is a highly flexible C++ MC simulation toolkit, developed at CERN [79]. Initially designed for high-energy applications (up to 100 TeV) and now capable of simulating down to a few electron volts. Its main feature is the ability to simulate the behavior of various particles, including exotic ones. GEANT4 offers an extensive range of models and processes, which need to be carefully selected depending on the specific use case. These models are categorized into data-driven, parameterized, and theory-based approaches. Although data-driven models are generally the preferred option, many models use a combination of all three approaches depending on the availability of data [80]. GEANT4 has been widely validated in various medical applications, including photon and electron physics in radiotherapy, as well as the electromagnetic and nuclear interactions for proton and carbon-ion therapy [81].

GATE (Geant4 Application for Emission Tomography) is an advanced, open-source software developed by the OpenGate collaboration [82]. Designed to facilitate medical imaging and radiotherapy simulations, it extends GEANT4's capabilities for use in medical applications, particularly for Positron Emission Tomography (PET), Single Photon Emission Computed Tomography (SPECT), and Computed Tomography (CT). Originally initiated in 2001 and released in 2004 [83], GATE incorporates specific features to

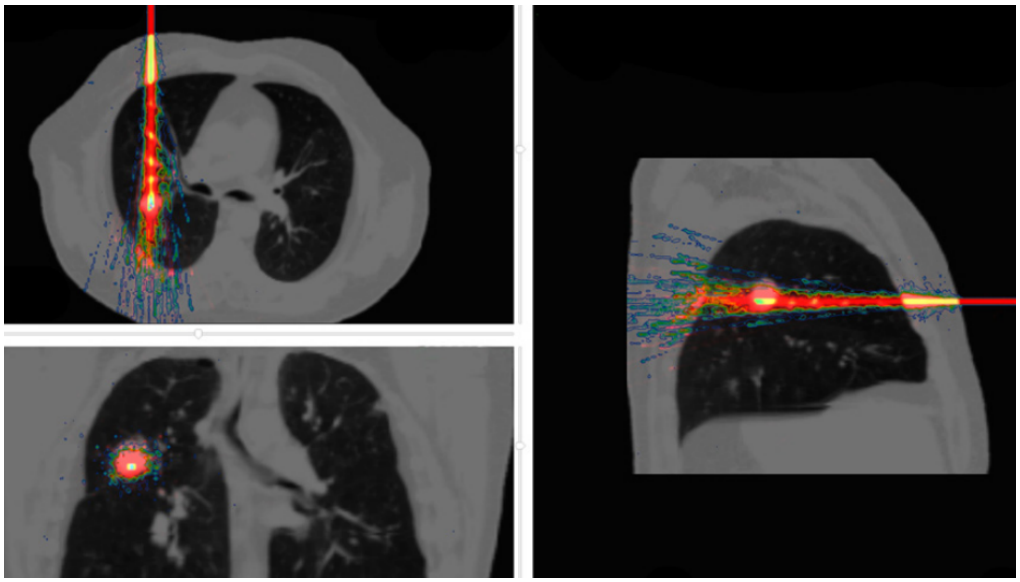


Figure 2.3: Dose deposited by a carbon ion beam inside a CT image of a thorax. The colour scale is a warm metal scale, with high values (white) corresponding to high-dose deposit and low values (blue) corresponding to low-dose deposit[75].

optimize GEANT4 for medical applications, including the management of time within simulations and the ability to handle complex geometries [84]. Its ease of use is enhanced by a macro-based scripting language, which simplifies the setup and control of experimental configurations, making it accessible to researchers without extensive programming experience [75].

In radiotherapy, GATE is used for simulating treatment procedures and radiation dose calculations, benefiting from the continuous advancements of GEANT4 in particle tracking and physical process modeling. Simulations for photon, electron, proton, and carbon-ion therapy are supported, allowing researchers to create detailed models of the treatment delivery, improving the accuracy of dose distribution and ultimately patient outcomes.

While faster MC codes, are available and more suitable for specific tasks like proton treatment planning [85], GEANT4's strength lies in its versatility. It can handle a wide variety of simulations, including imaging, dosimetry, and micro-dosimetry applications, and is widely regarded as a reference code for validation purposes [86]. Having a unified platform like GEANT4 for multiple medical physics applications is particularly advantageous given the growing complexity of radiotherapy treatments, especially image-guided techniques, which require precise dose calculations.

Additionally, GEANT4's capability to combine imaging and treatment simulations offers unique opportunities in the field of radiation oncology, facilitating research into new treatment modalities and contributing to improvements in patient safety and treatment efficacy.

Chapter 3

Simulation setup

In this chapter the tools employed and the methodologies used in this thesis will be discussed. The aim is to obtain a CBCT image of the head-and-neck region of the ICRP110 phantom [87]. First the MC code used will be described, detailing its evolution and adaptations. Next, an overview of the server where the simulations were executed will be provided. Finally, the software used to convert the various angular projections into DICOM files will be described.

3.1 MC code

In order to simulate a CBCT using a Monte Carlo code, a conical photon beam source has been implemented in front of the ICRP110 phantom shown in Fig.3.1. Moreover, a silicon Flat Panel detector was placed on the opposite side. During the simulations, particles are emitted from the source, they will interact with the phantom and subsequently will be detected by the Flat Panel detector. To reconstruct the CT image, variuos angles need to be simulated, to such purpose, the phantom was rotated around its axis at some fixed angles, maintaining the source and detector in their original positions. For these simulations it was chosen to use GATE at first, and than Geant4.

3.1.1 GATE

GATE is a toolkit based on Geant4, designed for the easy development of simulations in the field of medical imaging. It provides a template of a predefined geometry to simulate a scanner. For instance, in the "cylindricalPET" scanner system, the geometrical volumes containing crystals are grouped in matrices, themselves assembled in submodules and modules. At the top level of this structure, the sectors composed of modules are repeated on a cylindrical surface to build up the whole device. Thus, a family of PET scanners obeying this structure can be described using this system, composed of volumes called rsectors, modules, submodules, crystal and finaly (crystal) layer. Although it is possible to use GATE without employing these systems, in such cases, one cannot retrieve the hit information between particles and detectors provided by the digitizer shwon in Fig.3.3,

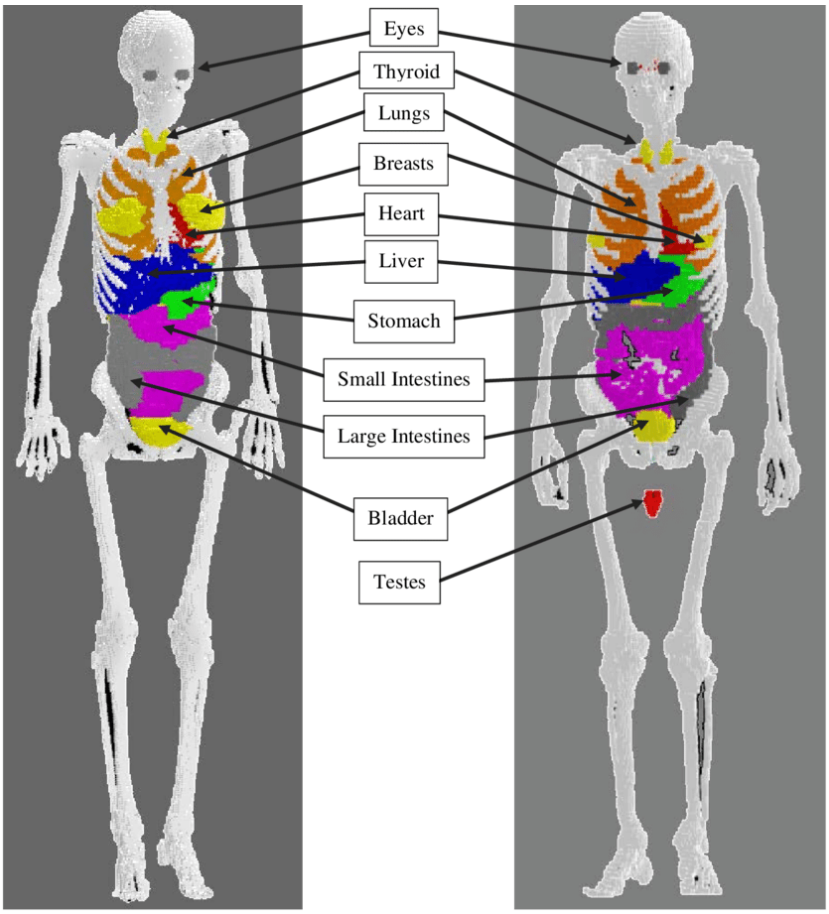


Figure 3.1: 3D rendering of whole body ICRP110 Reference Female (left) and Reference Male (right) voxel phantoms as modelled in the Geant4 application ICRP110Phantoms, in which skin, muscle, cartilage and adipose tissue are not visualised[88].

as these are only saved for volumes declared as "Sensitive Detectors", which exist only if connected to a system [90]. The system utilized was the "CTscanner", which is composed of three levels, all consisting of boxes that can be nested within one another: the "module", the "cluster" and the "pixel" the main component that allows for storing the projection image at each angle of the CT in a 32-bit matrix of size number of pixels x by number of pixels y, whose content corresponds to the number of "counts per pixel acquisition." "In the complete simulation, the modules, the clusters, and the pixels are user defined. All volumes are created by Geant4 and the digitalization can be made at the pixel level (level 3).

Thus, the geometry utilized is composed of the "World" which contains a box representing the "CTscanner" of dimensions 10x10x0.05 cm. This box, in turn, contains a module of the same dimensions, which houses a cluster that covers it entirely. The cluster finally contains pixels which are also boxes made of silicon with dimensions of 0.1 x 0.1 x 0.05 cm, repeated in a 100 x 100 matrix along the x and y axes, and which act as the Sensitive

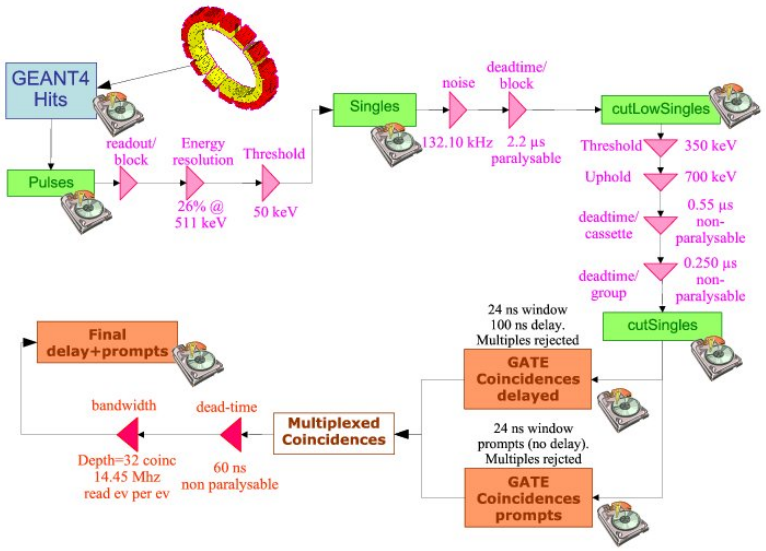


Figure 3.2: Readout scheme produced by the acquisition model of the ECAT HRRT scanner. The disk icons represent the data written to the GATE output files[89].

Detector.

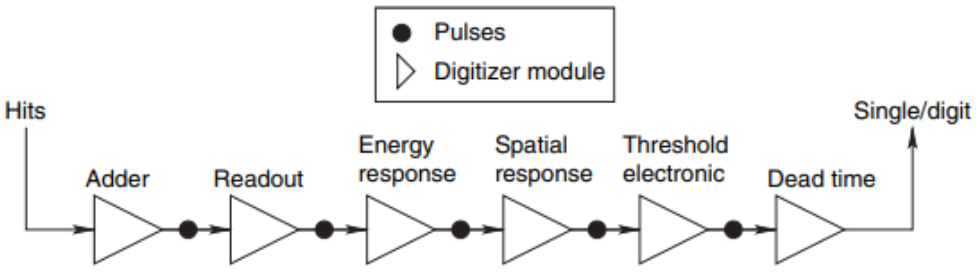


Figure 3.3: The digitizer is organized as a chain of several modules that processes the hits to yield a single, which represents a physical observable.[91].

Digitizer Module	**Short description**
Adder	Adds all *Hits* in one crystal
Optical adder	Adds all *Hits* generated by optical photons
Readout	Models readout by an individual photo-detector
Energy resolution	Applies Gaussian blurring on energy
Time resolution	Applies Gaussian blurring on time
Spatial resolution	Applies Gaussian blurring on 3D position
Energy framing	Selects an energy window
Efficiency	Applies detection system sensitivity
Pile-up	Models event pile-up
Dead time	Models detection system's dead time
Noise	Adds background events in a generic way for any kind of noise source
Adder Compton	Specific adder for Compton kinematics
Merger	Merges *Singles* from different *Sensitive Detectors*

The Digitizer represents a significant difference between GATE and Geant4: The purpose of the digitizer module is to simulate the behaviour of the scanner detectors and signal processing chain. GATE uses Geant4 to generate particles and transport them through the materials. The information generated during this process is used by GATE to simulate the detector pulses (digits), which correspond to the observed data. The digitizer represents the series of steps and filters that make up this process. The typical data-flow for an event is as follows:

A particle is generated, with its parameters, such as initial type, time, momentum, and energy. An elementary trajectory step is applied. A step corresponds to the trajectory of a particle between discrete interactions (i.e. photoelectric, Compton, pair production, etc). During a step, the changes to a particle's energy and momentum are calculated. The length of a step depends upon the nature of the interaction, the type of particle and material, etc. The calculation of the step length is complex and is mentioned here only briefly.

If a step occurs within a volume corresponding to a sensitive detector, the interaction information between the particle and the material is stored. For example, this information may include the deposited energy, the momentum before and after the interaction, the name of the volume where the interaction occurred, etc. This set of information is referred to as a Hit.

Steps 2 and 3 are repeated until the energy of the particle becomes lower than a predefined value, or the particle position goes outside the predefined limits. The entire series of steps form a simulated trajectory of a particle, that is called a Track in Geant4.

The amount of energy deposited in a crystal is filtered by the digitizer module. The output from the digitizer corresponds to the signal after it has been processed by the Front End Electronics (FEE). Generally, the FEE is made of several processing units, working in a serial and/or in parallel. This process of transforming the energy of a Hit

into the final digital value is called Digitization and is performed by the GATE digitizer. Each processing unit in the FEE is represented in GATE by a corresponding digitizer module. The final value obtained after filtering by a set of these modules is called a Single. Singles can be saved as output. Each transient value, between two modules, is called a Digi.

This process is repeated for each event in the simulation in order to produce one or more sets of Singles. These Singles can be stored into an output file (as a ROOT tree, for example).

In case of PET systems, a second processing stage can be inserted to sort the Singles list for coincidences. To do this, the algorithm searches in this list for a set of Singles that are detected within a given time interval (the so called 'coincident events').

Finally, the coincidence data may be filtered-out to mimic any possible data loss which could occur in the coincidence logical circuit or during the data transportation. As for the Singles, the processing is performed by specifying a list of generic modules to apply to the coincidence data flow. The Singles Digitizer is organized as a chain of digitizer modules that begins with the hit and ends with the single which represents the physical observable seen from the detector. The output from a digitizer module corresponds to the signal after it has been processed by the Front End Electronics (FEE).

Fig 3.4 illustrates the actions of both the adder and readout modules. The adder module transforms the hits into a pulse in each individual volume and then the readout module sums a group of these pulses into a single pulse at the level of depth as defined by the user for the winner-takes-all policy. The other modules are self explanatory.

GATE microCBCT of cylindrical phantom

The first simulation on GATE was conducted by analyzing the example of the small animals microCBCT simulation created by the comunity of OpenGATE collaboration, GateContrib [93] that is based on a cylindrical phantom with a radius of 8 mm and a height of 10 mm, filled with water. Inside the phantom, four spheres made of aluminum, PVC, "spineBone", and glass were placed on two planes at different heights, aligned along two diameters perpendicular to each other. The detector used was a cluster of 100 x 100 pixels, each measuring 0.5 x 0.5 x 1 mm, made of silicon. The source was a point gamma source emitting a conical beam with a half-angle of 6.8 degrees and energies ranging from 10 to 40 keV. The distance between the source and the cylinder's rotation axis was 15 cm, while the distance between the source and the detector was 30 cm. With this dimensions, the setup qualifies as a microCBCT.

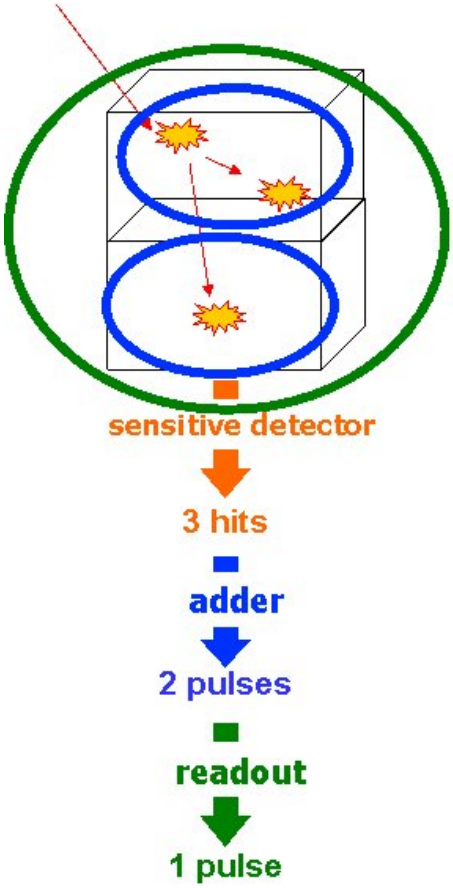


Figure 3.4: Actions of the adder and readout modules.[92].

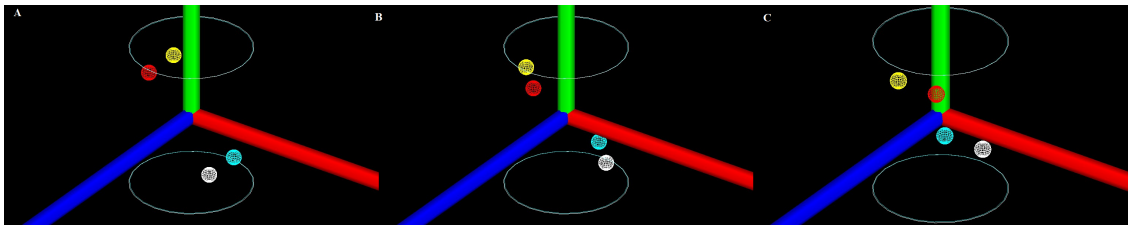


Figure 3.5: Cylindrical phantom rotated at different angles A)0°, B)45°, C)90°.

The source emitted 5×10^6 photons, and the detector recorded the results processed by a digitizer consisting of an adder, readout, and an energy threshold set to 10 keV. These results as can be seen in figure 3.2 were saved in a ROOT file as the default output of GATE, structured as follows: latest_event_ID:1, total_nb_primaries:1, pet_data:1, Hits:1, OpticalData:1, Singles:1; Singles is composed of:

`runID`	`globalPosX`
`eventID`	`globalPosY`
`sourceID`	`globalPosZ`
`sourcePosX`	`gantryID`
`sourcePosY`	`moduleID`
`sourcePosZ`	`clusterID`
`time`	`pixelID`
`energy`	`unused4ID`
`comptonPhantom`	`axialPos`
`comptonCrystal`	`rotationAngle`
`RayleighPhantom`	`comptVolName`
`RayleighCrystal`	`RayleighVolName`
`unused5ID`	`volumeID`

Due to issues with the GATE version used(GATEv.9.3), the rotation could not be automated within the program. Therefore, is implemented a script to run 360 different simulations, each with a different rotation angle of the phantom. This approach allowed for the acquisition of 360 distinct root files that are the projections at different angles as seen in figure 3.5.

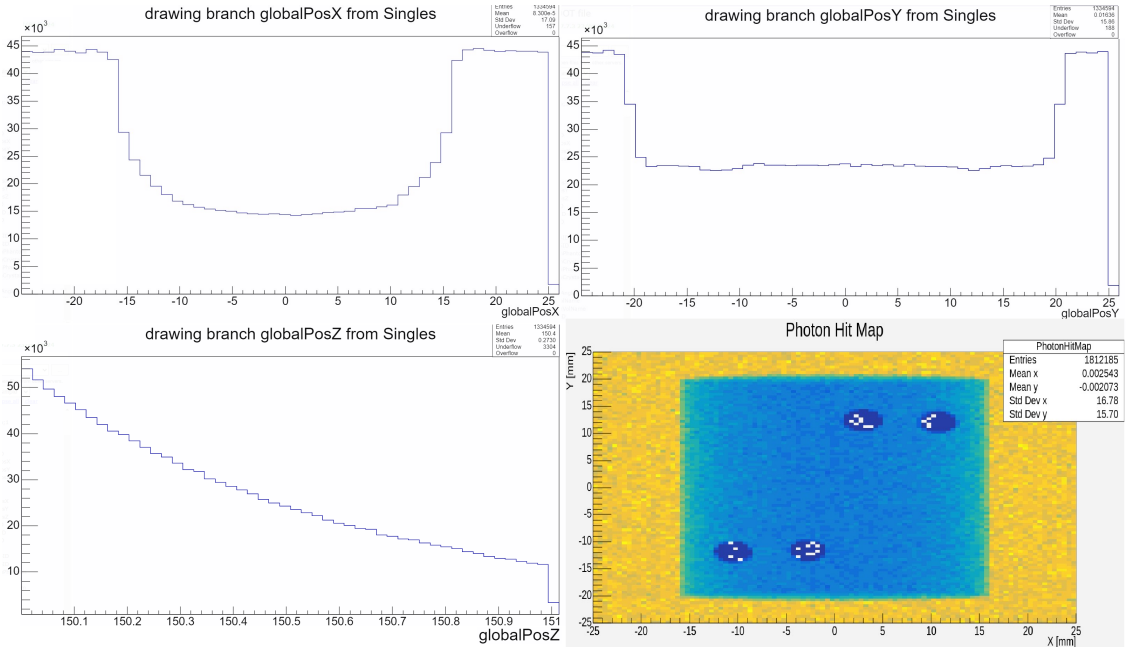


Figure 3.6: `globalPosX` `globalPosY` and `globalPosZ` recorded by the detector. In the lower right corner the reconstructed 2-D histogram

The files used are `globalPosX`, `globalPosY`, and `globalPosZ` plotted in figure 3.6, which record the spatial coordinates of each photon hit with the detector in an histogram.

globalPosX and globalPosY can be rearranged in a Root TH2F, than saved in a ".txt"

file as a matrix of numbers through code in Appendix A.1 and then converted in jpg images through code in Appendix B.1 that can be putted into OSCaR to do the DICOM volumetric reconstruction seen in Figure 3.7

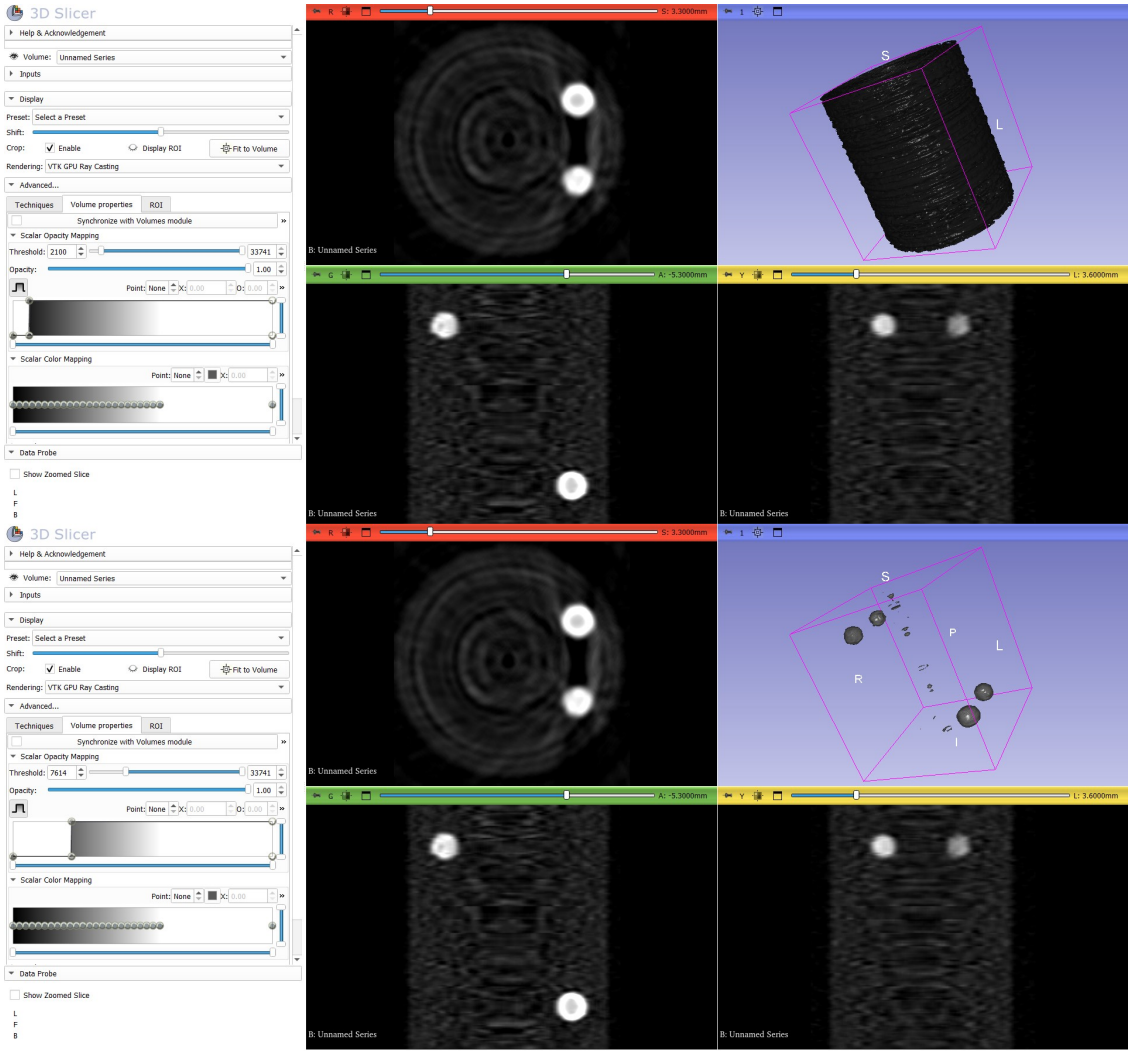


Figure 3.7: DICOM data of the reconstruction of the cylindrical phantom viewed with 3DSlicer, with different scalar opacity treshold, up left there is the Axial view, up righ there is the volumetric reconstruction, lower left the Coronal view and lower righ the Saggital one.

GATE CBCT of big cylindrical phantom

The aim of the second simulation was to scale up the geometry, transitioning from microCT to CBCT, and replacing the cylinder's diameter and height with the maximum dimensions of the human body, as referenced in ICRP110[94] 3.1. Thus, the cylinder's radius was set to 297.03 mm, and the height to 1776 mm, while still containing the four

spheres.

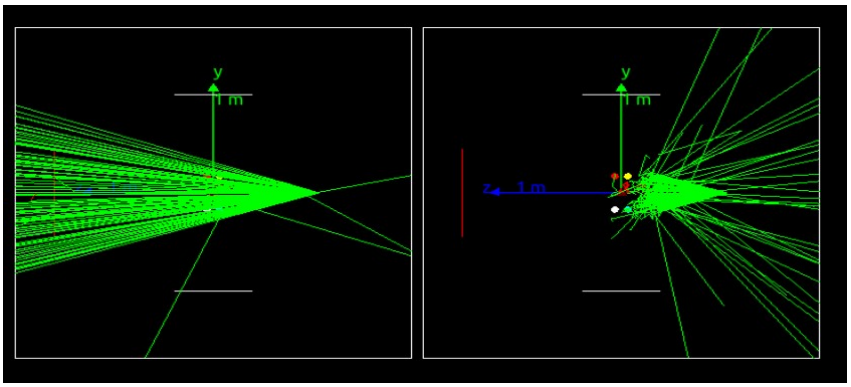


Figure 3.8: Simulation in gate with the enlarged geometry, in the left made of air, in the right simulation made of water

However, this presented significant challenges. With such a large thickness of water, a substantially greater number of photons is required to achieve statistically acceptable results, necessitating a much higher power output. From the image 3.8 it can be seen the percentage of photons that passes through the phantom. This is the primary reason for choosing Geant4 that can be run in multithread mode; also the output of the GATE simulation for a problem in the specific version used had to be only in the root file previously presented and this has slowed down a lot the simulation time.

3.1.2 Geant4

The Geant4 version used is the Geant4-11.2.2. As schematized in Figure 3.9 the user must describe the experimental setup, provide the primary particle input, choose the particles to be simulated and the physics model to be used and the precision of the simulation.
To do so there are four mandatory user classes: G4VUserDetectorConstruction and G4VUserPhysicsList that are invoked at the initialisation; G4VUserActionInitialization and G4VUserPrimaryGeneratorAction that is invoked during the execution loop. There are also optional user classes that permitt to the user to interact with G4 kernel, visualize and produce histograms for example.

To build a simulation the user must write the main program were there is the construction of the G4RunManager and the notification of the mandatory user classes, at the end the G4RunManager must be deleted. To define the geometry the user has to use an inerith class from G4VUserDetectorConstruction and register it to the Run Manager. To define the physics processes the user has to use an inerith class from G4VUserPhysicsList if he wants to manipualate it he has to define all necessary particles, processes and particle production threshold (in terms of range) through pure virtual methods: ConstructParticles(), ConstructProcesses() and SetCuts()

The user has also to instantiate at least the primary generation that is invoked in sequential mode and in MT mode by all workers, is inherit form G4VUserPrimaryGenerator via the ActionInitialization (only in MT mode) and be registered to the Run Manager. Must be implemented the method "GeneratePrimaries(G4Event)" that is called dureing the event loop to generate the primary particles.

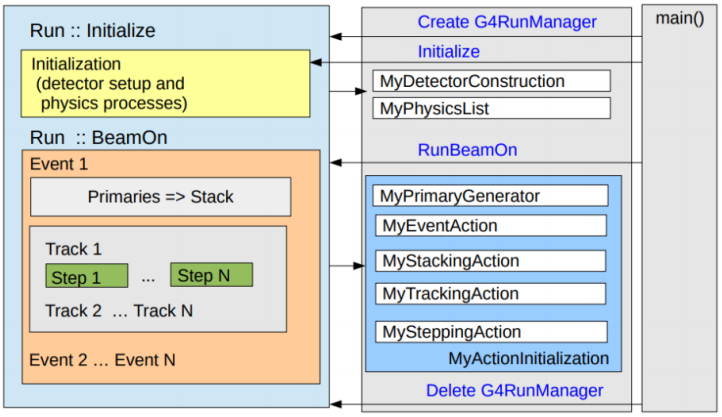


Figure 3.9: The general recipe of a Geant4 simulation: mandatory user classes MyDetectorConstruction and MyPhysicsList as initialization classes, MyPrimaryGeneratorAction and MyActionInitialization as Action classes; and also optional derived user action classes. On the left can be seen also the division of the simulation in Run, Events, Tracks and Steps [95].

Geometry

The code presented is inspired to the Geant4 Advanced example of a voxelized phantom, which is a three dimensional representation of the human body divided into smaller volumetric elements, or voxels. These voxels are used to simulate the interaction of radiation with various tissues within the human body. The particular code relates to the ICRP (International Commission on Radiological Protection) 110 phantom model [87], and it provides functionalities for handling both male and female phantoms with customizable sections (head, trunk, or full body). In this section, is provided a detailed and thorough explanation of the C++ code for the class ICRP110PhantomConstruction, which is responsible for constructing the phantom geometry [96].

The class constructor, ICRP110PhantomConstruction(), initializes several key variables that control the construction of the phantom:

- `fMotherVolume`: Stores the mother volume, which is the logical volume that contains all other volumes.
- `fPhantomContainer`: Stores the logical volume that contains the voxelized phantom.
- `fNVoxelX, fNVoxelY, fNVoxelZ`: The number of voxels in the X, Y, and Z directions, respectively.

- `fVoxelHalfDimX, fVoxelHalfDimY, fVoxelHalfDimZ`: Half-dimensions of the voxels along the X, Y, and Z axes, respectively.
- `fMinX, fMaxX, fMinY, fMaxY, fMinZ, fMaxZ`: These variables store the minimum and maximum bounds of the voxelized geometry along each axis.
- `fMateIDs`: A pointer to an array that stores the material IDs for each voxel.
- `fMaterial Female` and `fMaterial Male`: These objects manage the materials specific to female and male phantoms, respectively.
- `fMessenger`: An object that allows interactive communication between the user and the simulation.
- `fSex` and `fSection`: These variables store the gender ("male" or "female") and the section ("head", "trunk", or "full") of the phantom to be built.

The constructor initializes the messenger and sets the default values for the phantom sex and section. The default sex is set to "male", while the default section is set to "full", meaning that unless the user specifies otherwise, a male full-body phantom will be built.

The destructor ensures that memory allocated for the messenger, female materials, and male materials is properly released when the object is destroyed.

The method `Construct()` begins by defining the materials that will be used in the simulation. The primary material defined here is air, which is a mixture of nitrogen and oxygen. These elements are created using the Geant4 class `G4Element`, and the material air is created by combining these elements in the appropriate ratio using the `G4Material` class. The density of air is set to $0.001 g/cm^3$, and the code adds nitrogen and oxygen in a ratio of 80% and 20%, respectively.

```
G4double A; G4double Z; G4double d;

A = 14.01g/mole; auto elN = new G4Element("Nitrogen","N",Z = 7.,A); A =
↪  16.00g/mole; auto elO = new G4Element("Oxygen","O",Z = 8.,A);
```

Next, the NIST database is used to retrieve the material for silicon, which will be used for the detector.

```
G4NistManager* nist = G4NistManager::Instance(); G4Material* silicon =
↪  nist->FindOrBuildMaterial("G4_Si");
```

The world volume is the outermost volume in the simulation, which contains all other volumes. In this case, the world is defined as a cubic box with a side length of 3 meters, this dimension is choosen to be able to put source and detector at a distance of 161cm and to put the ICRP110 phantom center in y=-72.5cm so the source cone beam interacts with the head and neck section. The material used for the world is air, defined earlier. This world volume is associated with a logical volume (logicWorld) and a physical volume (fMotherVolume), with the latter being placed in the simulation as the root volume.

```
G4double worldSize = 3.m ; G4Box world = new G4Box("world", worldSize,
↪  worldSize, worldSize); auto logicWorld = new G4LogicalVolume(world, matAir,
↪  "logicalWorld", nullptr, nullptr,nullptr); fMotherVolume = new
↪  G4PVPlacement(nullptr,G4ThreeVector(),"physicalWorld",logicWorld,nullptr,false,0);
```

A detector is also defined within the world volume. The detector is modeled as a silicon box with dimensions of 70 cm x 70 cm x 1 cm. This detector is placed at a position offset along the Z-axis (80.5 cm from the origin). This volume is used to simulate interactions between particles and the detector.

```
G4double detector_sizeXY = 70. * cm; G4double detector_sizeZ = 1.0 * cm; G4Box*
↪  solidDetector = new G4Box("Detector", detector_sizeXY / 2, detector_sizeXY /
↪  2, detector_sizeZ / 2); G4LogicalVolume* logicDetector = new
↪  G4LogicalVolume(solidDetector, silicon, "Detector"); new G4PVPlacement(0,
↪  G4ThreeVector(0, 0, 80.5 * cm), logicDetector, "Detector", logicWorld,
↪  false, 0);
```

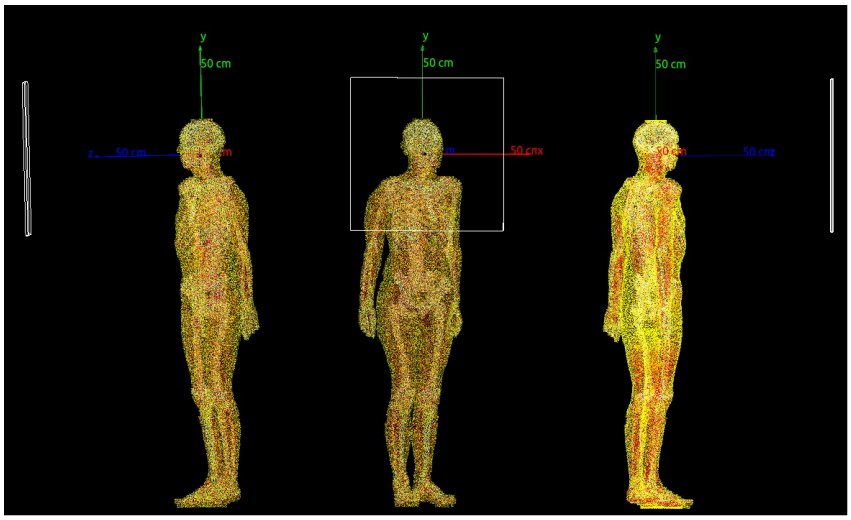


Figure 3.10: positioning of the phantom in space, with respect to the Detector.

The voxelized phantom is built by first defining a container volume (fContainersolid), which is a box that encloses all the voxels. The size of this container is determined by the number of voxels along each axis and the half-dimensions of each voxel. This container is then placed in the world volume in the center but at y=-72.5cm so the source cone beam interacts with the head and neck section; it is placed also with a specified rotation matrix that will change rotating the phantom in order to obtain different angles projections, this modification has been automatized trough the code in Appendix B.2. The container is then further subdivided into smaller regions (replicas) along the Y, X, and Z axes, where each voxel is placed.

The code uses the Geant4 feature of parameterized placement (`G4PVParameterised`) to efficiently place voxels in the container, with the material of each voxel being determined dynamically by the parameterization class `ICRP110PhantomNestedParameterisation`.

```
G4Box* fContainer_solid = new G4Box("phantomContainer",
↪  fNVoxelXfVoxelHalfDimXmm, fNVoxelYfVoxelHalfDimYmm,
↪  fNVoxelZfVoxelHalfDimZmm); auto fContainer_logic = new G4LogicalVolume(
↪  fContainer_solid, matAir, "phantomContainer", nullptr, nullptr, nullptr);
↪  fPhantomContainer = new G4PVPlacement(rotCylinder,
↪  G4ThreeVector(0,-72.5*cm,0), fContainer_logic, "phantomContainer",
↪  logicWorld, false, 1);
```

The voxelized phantom is then sliced along the Y, X, and Z axes to create smaller regions. Each region is further divided into voxels, where the material for each voxel is assigned based on the input data files as can be seen in figure 3.10.

The method `ReadPhantomData()` is responsible for reading the data necessary to construct the voxelized phantom. Depending on the selected sex and section of the phantom, the method opens different data files (e.g., FemaleData.dat or MaleData.dat) that contain the geometric and material information for each voxel.

Each file specifies the number of voxels, voxel dimensions, and the material IDs for each voxel. The `ReadPhantomDataFile()` function processes each slice of the phantom and maps the organ IDs (from the input file) to the appropriate material IDs.

The functions `SetPhantomSex()` and `SetPhantomSection()` allow the user to customize the phantom's sex and section interactively. These functions modify the `fSex` and `fSection` variables, respectively, and print relevant messages to the user.

Other components of the Geant4 simulation

With the transition to Geant4, a significant improvement in performance was observed. This is because, the detector was geometrically treated as a single large silicon flat panel, surpassing the physical division into pixels. The information is not stored through a digitizer but using a custom-written code.

The code in appendixA.2.2 invokes ICRP110PhantomPrimaryGeneratorAction to initialize the source and ICRP110PhantomSteppingAction, which, in conjunction with AnalysisManager exposed in AppendixA.2.4, creates a 2D histogram (TH2F) using ROOT libraries previously imported via CMakeList [96].

The histogram is populated through the "ICRP110PhantomSteppingAction" that can be seen in AppendixA.2.3

The source was defined in the file ICRP110PhantomPrimaryGeneratorAction that can be seen in AppendixA.2.5. It is a fParticleGun modeled as a cone with its vertex at z= −80.5cm, an opening angle of 16 degrees, and energy uniformly distributed between 10 and 140 keV.

To complete the mandatory classes cited in Figure 3.9, a custom `PhysicsList` has been defined that can be seen in appendixA.2.6 where are configured the particles `Gamma`, `Electron`, and `Positron`. There is implemented their transportation processes and, for gamma particles, activated the photoelectric effect and Compton scattering processes.

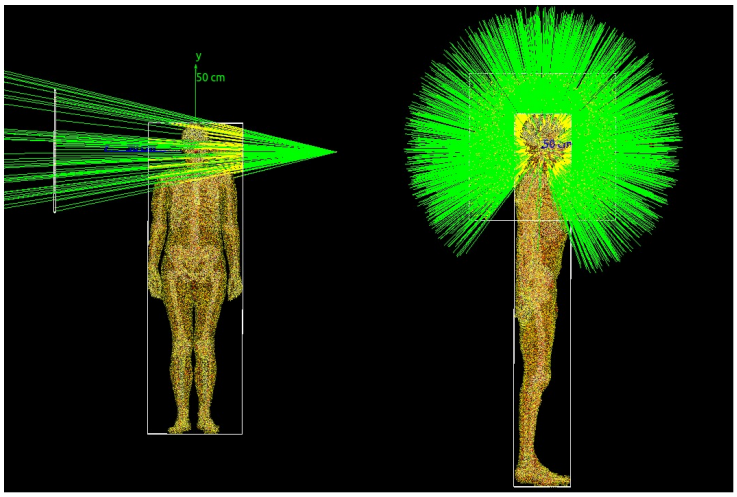


Figure 3.11: Cone beam source of photons.

For electrons, there is enabled multiple scattering and ionization processes. Additionally, there is defined a cut-off value of 0.3 mm.

The processes for photons have been customized as can be seen in appendix A.2.7 and A.2.8. To speed up the simulation and avoid the complications of defining an "Anti-scatter grid" 1.17 which would be complex due to the conical beam and would also remove particles with the correct direction, thereby reducing the statistics, has been implemented a physics where if a photon undergoes one of the aforementioned interactions, secondary particles are generated if expected, but both the primary particle and the secondary particles immediately cease to exist at the location where the interaction occurs Is the same for Compton Scattering. This can be seen in Figure 3.11 where photons are genereted in a conical source and don't have any scatter along their track.

Multithreading and Server

Another significant advantage of Geant4 is its native support for multithreading.Initially worked with servers, starting with one provided through a collaboration between the University of Pavia and GitHub Education. This collaboration allowed access to a small server via the DigitalOcean service, enabling necessary adjustments to the Geant4 code in preparation for its subsequent use on a more powerful cluster provided by the Department of Physics.: `mellon.pv.infn.it`. So in the main there was specified that there are used 48 Threads A.2.1

A big problem that occurred was related to the different behavior of the simulation results that occurred using more than 1 threads, indeed using more threads photons interacts with the Phantomcontainer and a box can be seen in the TH2F. After a long debugging of the code, searching on the web conduct to the Geant4 forum [97] where a user described that in some versions of Geant4 there was a problem between Multithread and energy deposited in the ICRP110 phantom. The problem has been resolved using the newest

version of Geant4 (Geant4-11.1.3).

The cluster of INFN is equipped with Intel(R) Xeon(R) CPU E5-2680 v3 processors, each operating at 2.50 GHz. This CPU model has 12 physical cores, and each core can run two threads simultaneously thanks to Hyper-Threading technology, providing a total of 24 threads per CPU. Each machine is configured with two CPUs, yielding a total of 48 threads per machine and 64 GB of total RAM. There was granted access to work on 4 machines in parallel.

After more than 70 hours, using 4 nodes at the maximum performance, running with 10^8 prymary photons per 360 different angles the results are 360 ROOT TH2F files, rappresenting all the different projections at the different angles; an example is provided in the figure 3.12; it can be seen that the statistics is not so good and this is due to the small number of primary particles used.

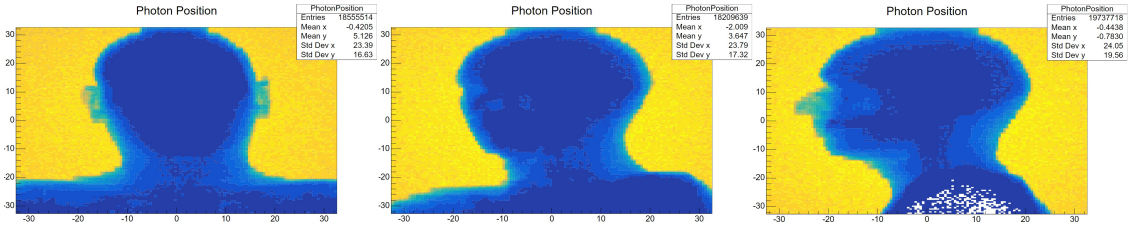


Figure 3.12: TH2F projections of head and neck sector of ICRP110 phantom obtained with my Geant4 simulation using 10^8 prymary photons. From left to right with a rotation angle of the phantom of 0°, 45° and 90°

3.2 DICOM reconstruction

To do the reconstruction it is used a program to convert the simulation output to jpg images and to create the csv file.

For Geant4 output B.2 will create a folder with 360 different images jpg of the different projections; an example can be seen in Figure 3.13.

With another python script the csv file has been constructed.

Then the DICOM files can be read for example with 3D Slicer (Slicer) that is a free and open source software package for image analysis [98]

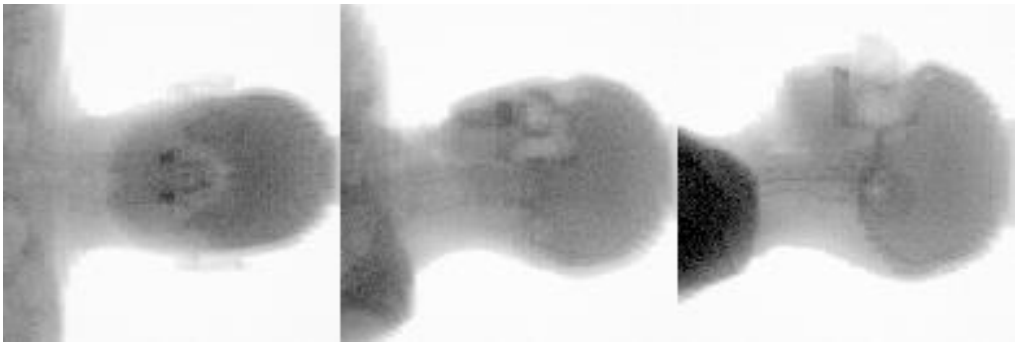


Figure 3.13: jpg projections from ROOT Geant4 simulation output; from left to right $0°, 45°$ and $90°$ rotation angle

Chapter 4

Results and Discussion

The DICOM files obtained from the OSCAR software, which correspond to the CBCT reconstruction of the ICRP110 phantom's head&neck area, were opened with 3D Slicer 5.6.2 and can be seen in figure4.1 It can be noted tat there are circular artifacts caused by the CBCT. There is noise because the CBCT reconstruction is good only in the central layer (i.e. the one which fulfills the Tuy condition). Moreover, the small number of particles used have caused artifacts at the level of the shoulders. All this can be improved with two main changees: number of primary photons has to be increased (but this will increase exponentially the time of the simulation), and the cone angle hat to be decreased. If there is a focus in a specific small section of the phantom the tuy condition will be fulfilled. Furthermore, the number of photons that interacts become greater and, keeping the same time of simulation (more or less 70 hours with 48 x 4 threads), the noise will decrease.

The generation of *in-silico* CT images from humanoid phantoms is of particular interest in the context of artificial intelligence[1] (AI) for clinical settings. Generally, the datasets available to train *deep learning* models are composed of few data that rarely contain healthy patients. Having a system that enables the generation of volumetric diagnostic images would allow us not only to expand the datasets but also to improve the training of the algorithms.

At present, deep learning methods applied to clinical cases are largely confined to pre-clinical research. Nevertheless, the continuous improvement of these automatic methods may eventually lead to systematic collaboration between clinicians and AI, revolutionizing cancer treatment [101]. In the realm of cancer imaging, AI offers significant advantages to clinicians. Its primary applications include tumor detection, characterization, and monitoring, as well as the automatic segmentation of organs. AI can identify complex patterns in medical images and extract quantitative features—information that is often imperceptible to the human eye. This transforms image interpretation from a qualitative,

[1] AI is defined as "a system's ability to correctly interpret external data, learn from such data, and apply those learnings to achieve specific goals and tasks through flexible adaptation [99]

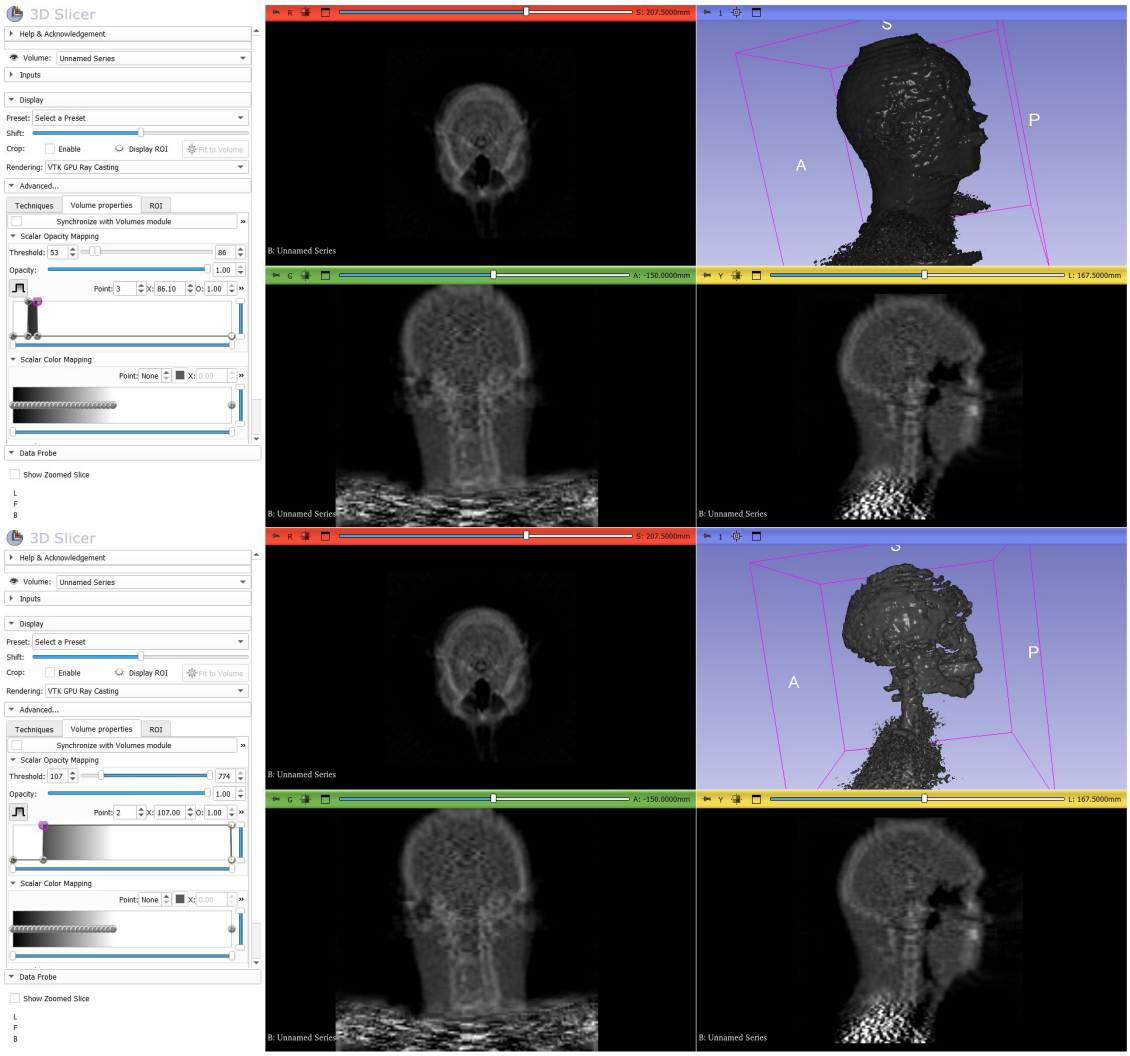


Figure 4.1: DICOM data OSCaR output of the reconstruction of the ICRP110 phantom trough Geant4 based CBCT simulation; viewed with 3DSlicer, with different scalar opacity treshold, up left there is the Axial view, up righ there is the volumetric reconstruction, lower left the Coronal view and lower righ the Saggital one.

subjective process into one that is measurable and reproducible, assisting clinicians in making more informed decisions [101].

Image segmentation is another key AI application, involving the extraction of quantitative information about tumor lesions (Region of Interest, ROI) or organs regarding their volume, morphology, and texture patterns. This information is crucial for both diagnostic purposes and the development of radiation treatment planning systems (TPS) [102]. In radiotherapy or hadrontherapy, the TPS process begins with the segmentation (contouring) of the target volume (the tumor) and organs at risk (OAR) using CT, MRI, and PET scans. Precise delineation of margins is essential, as it forms the foundation for radiation

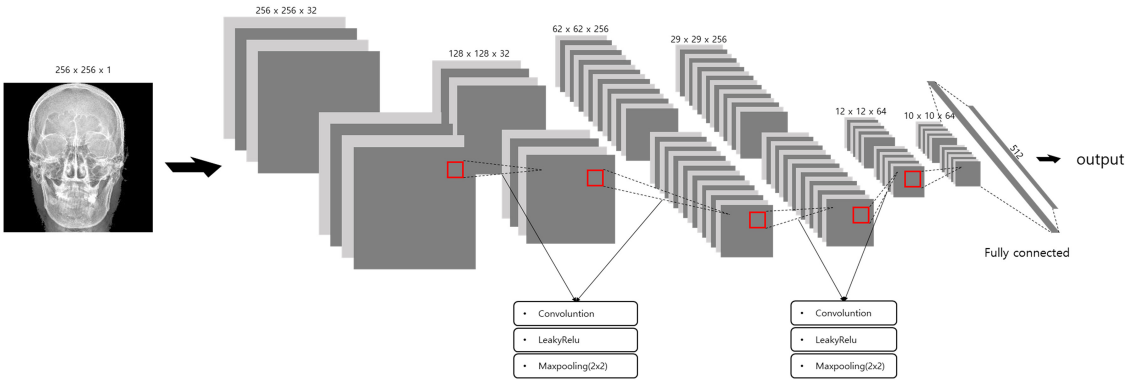


Figure 4.2: Diagram shows schematic structure of convolutional neural network. The skull image is of one of the authors of the article[100].

beam management to reduce doses to healthy tissues while maintaining therapeutic doses to the tumor [103].

Traditionally, contouring of ROIs and OARs is done manually by experienced radiologists, a practice considered the gold standard due to its perceived accuracy. However, this is a subjective task influenced by:

- Intra-observer variability, which occurs when the same observer contours the same target multiple times, leading to discrepancies between the contours.
- Inter-observer variability, which reflects differences in contours drawn by different observers on the same target.

Such variability is further exacerbated by the low quality of clinical images. For example, in low-dose CT scans, the contrast between soft tissues is often poor, increasing the likelihood of errors due to intra-/inter-observer variability [104][105]. Moreover, manual contouring is time-consuming and tedious for radiologists. An automatic segmentation system could help alleviate these issues.

Deep learning can significantly aid automatic segmentation by reducing the time required for the process and eliminating human-induced variability[106]. Anyhow, these methods require large training sets. Existing public repositories such as The Cancer Imaging Archive [107] contain many but numerically limited clinical datasets. Thus, a significant challenge lies in the limited number of patients and the lack of images of healthy individuals without tumors or lesions. This thesis work originated from the idea of expanding the dataset of images of healthy patients available online by creating synthetic Cone Beam CT scans of phantoms using Monte Carlo methods.

Chapter 5

Conclusion & Perspectives

In this thesis, BNCT has been discussed, including its operating principles and the selectivity of the therapy. In particular, this work focused on the *in-silico* reconstruction of a diagnostic image of the head-neck region, since it is one of the primary tumoral targets of this therapy.

This dissertation focuses on the reconstruction of CT images from MC simulations. By starting from the production of X-rays, passing through the generation of projection images and arriving at the volumetric reconstruction of CBCT through the OSCaR software. The thesis also explores emphasize the specific MC methods used for CT simulations, namely GATE and Geant4.

Special attention is given to the code, starting with the initial work using GATE and the performance reasons that made the transition to Geant4 requiring the definition of geometry (and therefore explained the use of ICRP110), source, physics and data storage through ROOT. The process from simulation to DICOM file generation using the OSCaR software is also explained. In the results, a volumetric reconstruction of the head-neck area of the phantom can be seen, although with possible improvements. In the future, this work could be improved, for example by changing the tomography method used, or perhaps more importantly, by changing the phantom used, in fact the tetrahedral ICRP145 [108] phantom could be used, which is more defined and could be modelled so as to expand the database even further with images of healthy patients in different positions. One could also place a tumour in a specified position in the phantom and train the neural network to detect it.
In the future, these changes can be implemented to generate CT images from phantoms, which could be used for training AI algorithms. The code developed in this work has been made available on GitHub, providing a foundation for further advancements in both research and clinical practice.

Appendix A

MC source code

In this appendix will be discussed the codes used to construct the 360 projections, one for each degree of rotation of the phantom, obtained using the MC GATE and Geant4 methods that can be found in the GitHub repository [96].

A.1 Gate

This software reads a ROOT file containing data from a tree (TTree) and creates a two-dimensional histogram representing the globalPosX and globalPosY projection of the photons detected during the first run. It then draws this histogram on a root TH2F and saves the histogram data in a text file (histogram_data.txt) as an array of numbers, each number corresponding to a cell in the 2D histogram [109]:

```
#include <iostream>
#include <cerrno>
#include <cstdlib>
#include <fstream>

#include "TApplication.h"
#include "TFile.h"
#include "TCanvas.h"
#include "TTree.h"
#include "TH2F.h"

using namespace std;

int main( int argc, char* argv[] )
{
    if( argc < 2 )
    {
        cerr << "arguments missing" << endl;
        cerr << "Usage : AnalyzeCT myFile.root " << endl;
        exit( EXIT_FAILURE );
```

```
}

// Store the root file name in 'fileName' variable
char* const FILENAME = argv[ 1 ];

TApplication app( "Application", &argc, argv );

// Create and initialize a canvas
TCanvas* canvas = new TCanvas( "Canvas BenchmarkCT", FILENAME, 200, 20,
↪  1000, 700 );
canvas->SetFillColor( 29 );
canvas->ToggleToolBar();
canvas->ToggleEventStatus();

// Open (check) and read the root file
TFile* file = new TFile( FILENAME );
if( !file->IsOpen() )
{
    cerr << "problem opening the root file : '" << FILENAME << "'" << endl;
    cerr << strerror( errno ) << endl;
    exit( EXIT_FAILURE );
}

// Take the single tree, where is the position, the energy and the runID
TTree* singlesTree = (TTree*)file->Get( "Singles" );
// Global Position in X, Y and Z
Float_t globalPosX, globalPosY;
singlesTree->SetBranchAddress( "globalPosX", &globalPosX );
singlesTree->SetBranchAddress( "globalPosY", &globalPosY );

// Number of entries in the single tree
Int_t entriesSingleTree = (Int_t)singlesTree->GetEntries();
cout << "Number of detected photons : " << entriesSingleTree << endl;

// Create histogram for the first run
// Define the bounds of the histogram
Double_t const PIXELSIZE = 0.5;
Int_t const RAW = 100;
Int_t const COLUMN = 100;
Double_t const RAW_BOUND = PIXELSIZE * RAW / 2;
Double_t const COLUMN_BOUND = PIXELSIZE * COLUMN / 2;
TH2F* run_0 = new TH2F( "runID = 0", "projection during the first run",
                        COLUMN, -COLUMN_BOUND, COLUMN_BOUND,
                        RAW, -RAW_BOUND, RAW_BOUND );

for( Int_t i = 0; i != entriesSingleTree; ++i )
{
    singlesTree->GetEntry( i );
    run_0->Fill( globalPosX, globalPosY );
}
```

```
    // Draw the histogram on the canvas
    canvas->cd();
    run_0->Draw( "COLZ" );
    // Apri un file di testo per la scrittura
        ofstream outputFile("histogram_data.txt");

// Verifica se il file è stato aperto correttamente
        if (!outputFile.is_open()) {
            cerr << "Errore nell'apertura del file di output." << endl;
            exit(EXIT_FAILURE);
        }

// Loop per estrarre i valori dei bin dall'istogramma e scriverli nel file di
↪  testo
        for (int i = 1; i <= COLUMN; ++i) {
            for (int j = 1; j <= RAW; ++j) {
                outputFile << run_0->GetBinContent(i, j) << " ";
            }
            outputFile << endl; // Vai a capo dopo ogni riga
        }

// Chiudi il file di testo
        outputFile.close();

    app.Run();

    return 0;
}
```

A.2 Geant4

In this appendix will be discussed all the subsections of the code used to construct the 360 projections, one for each degree of rotation of the ICRP110phantom, using the MC Geant4 method.

A.2.1 Main function

This subsection focus is the Main file of my simulation, which is the main file for an MC simulation on Geant4. First, the Run Manager has been createdwith G4RunManager to manage the execution of the simulation by setting up 48 threads to use the cluster to its full potential. Was also initialised the geometry with ICRP110PhantomConstruction using the model, then initialised the physics using a custom physics called SimplePhysicsList. A viewer is also initialised (G4VisManager); in addition, the ICRP110PhantomActionInitialisation is used to initialise particle generation, tracking and data collection. A.2.1

```
int main(int argc,char** argv)
{
 auto* runManager = G4RunManagerFactory::CreateRunManager();
 G4int nThreads = 48;
 runManager->SetNumberOfThreads(nThreads);
 // Set mandatory initialization classes
  auto userPhantom = new ICRP110PhantomConstruction();
  runManager -> SetUserInitialization(userPhantom);

  runManager -> SetUserInitialization(new SimplePhysicsList());

  G4VisManager* visManager = new G4VisExecutive;
  visManager->RegisterRunDurationUserVisAction
  ("phantom",new ICRP110PhantomVisAction(userPhantom));
  visManager -> Initialize();

  auto actions = new ICRP110PhantomActionInitialization();
  runManager -> SetUserInitialization(actions);

  G4UImanager* UImanager = G4UImanager::GetUIpointer();

  if (argc==1)   // Define UI session for interactive mode.
    {
      G4cout << " UI session starts ..." << G4endl;
      auto ui = new G4UIExecutive(argc, argv);
      UImanager -> ApplyCommand("/control/execute vis.mac");
      ui -> SessionStart();
      delete ui;
    }
  else           // Batch mode
    {
      G4String command = "/control/execute ";
```

```
      G4String fileName = argv[1];
      UImanager -> ApplyCommand(command+fileName);
    }

delete visManager;

delete runManager;

return 0;
}
```

A.2.2 Action Initialization

The method ICRP110PhantomActionInitialisation::Build() is used to configure actions for the Geant4 simulation. Through SetUserAction(new ICRP110PhantomPrimaryGeneratorAction) sets the primary action for particle generation, it is a class that defines how and where particles are generated at the start of the simulation. With SetUserAction(new ICRP110PhantomSteppingAction(fAnalysisManager)), we define what happens each time a particle takes a 'step' in the volume defined by the geometry.

```
void ICRP110PhantomActionInitialization::Build() const {
    SetUserAction(new ICRP110PhantomPrimaryGeneratorAction);
    SetUserAction(new ICRP110PhantomSteppingAction(fAnalysisManager));
```

A.2.3 Stepping Action

The constructor of the class (ICRP110PhantomSteppingAction) takes as its argument a pointer to an ICRP110PhantomAnalysisManager object, which is responsible for managing the data collected during the simulation. The UserSteppingAction method is called each time a particle takes a 'step' within the geometry It checks whether the particle taking the step is a photon (gamma), obtains the current position of the particle (G4ThreeVector position) and extracts its x, y and z co-ordinates, if the particle's z-coordinate is in the range between 80 cm and 81 cm, and if the kinetic energy at the end point of the step (postStepPoint) minus the kinetic energy at the start point of the step (preStepPoint) is greater than 0, the code calls the FillHistogram(x, y) method of the fAnalysisManager object:

```
#include "ICRP110PhantomSteppingAction.hh"
#include "G4Step.hh"
#include "G4Track.hh"
#include "G4SystemOfUnits.hh"

ICRP110PhantomSteppingAction::ICRP110PhantomSteppingAction(ICRP110PhantomAnalysisManager*
↪  analysisManager) :
    G4UserSteppingAction(), fAnalysisManager(analysisManager) {}

ICRP110PhantomSteppingAction::~ICRP110PhantomSteppingAction() {}

void ICRP110PhantomSteppingAction::UserSteppingAction(const G4Step* step) {
    G4Track* track = step->GetTrack();

    if (track->GetDefinition()->GetParticleName() == "gamma") {
        G4ThreeVector position = track->GetPosition();
        double x = position.x() / cm;
        double y = position.y() / cm;
        double z = position.z() / cm;
        G4StepPoint* preStepPoint = step->GetPreStepPoint();
        G4StepPoint* postStepPoint = step->GetPostStepPoint();

        if (z  > 80.  && z < 81. ){
           //if (x > -35.  && x < 35.  && y > -35.  && y < 35. ) {
           // Controlla se c'è stata una perdita di energia
            G4double energyLoss = preStepPoint->GetKineticEnergy() -
            ↪  postStepPoint->GetKineticEnergy();
            if (energyLoss > 0) {
            fAnalysisManager->FillHistogram(x, y);
           //}
           }
       }
   }
}
```

A.2.4 Analysis Manager

The constructor of the ICRP110PhantomAnalysisManager class creates a two-dimensional histogram (TH2F) to record the position of photons. The histogram is called 'PhotonPosition' and has 130 bins for both the x-axis and y-axis, with a range from -32.6 cm to 32.6 cm for both dimensions. This covers a square region of 65.2 cm x 65.2 cm. When the object is destroyed, a ROOT file is created (output_n.root, where n ranges from 0 to 360 and corresponds to the rotation angle of the phantom) in write mode (RECREATE), which deletes any previous content with the same name. The fHistogram is written to the ROOT file. The file is then closed, and TH2F is deleted from memory. The FillHistogram(double x, double y) method fills the fHistogram with the x and y coordinates passed as arguments:

```
#include "ICRP110PhantomAnalysisManager.hh"

ICRP110PhantomAnalysisManager::ICRP110PhantomAnalysisManager() {
    fHistogram = new TH2F("PhotonPosition", "Photon Position",
                          130, -32.6, 32.6, // binning for x
                          130, -32.6, 32.6); // binning for y
}

ICRP110PhantomAnalysisManager::~ICRP110PhantomAnalysisManager() {
    TFile* file = new TFile("output_330.root", "RECREATE");
    fHistogram->Write();
    file->Close();
    delete fHistogram;
}

void ICRP110PhantomAnalysisManager::FillHistogram(double x, double y) {
    fHistogram->Fill(x, y);
}
```

A.2.5 Source definition

Source definition, the constructor of the class ICRP110PhantomPrimaryGeneratorAction creates a G4ParticleGun object which generates the particles, which are set to 'gamma' by G4ParticleDefinition via G4ParticleTable. the initial position is set to (x,y,z)=(0,0,-50.5cm). Each time the ParticleGun is called, GeneratePrimaries takes energy from a uniform distribution between 10 and 40keV and a uniformly generated direction within a cone of 16 degrees around the z-axis.

```
#include "ICRP110PhantomPrimaryGeneratorAction.hh"
#include "G4ParticleGun.hh"
#include "G4ParticleTable.hh"
#include "G4ParticleDefinition.hh"
#include "G4SystemOfUnits.hh"
#include "Randomize.hh"
#include "G4GeneralParticleSource.hh"

ICRP110PhantomPrimaryGeneratorAction::ICRP110PhantomPrimaryGeneratorAction() {
    G4int n_particle = 1;
    fParticleGun = new G4ParticleGun(n_particle);

    G4ParticleTable* particleTable = G4ParticleTable::GetParticleTable();
    G4String particleName;
    G4ParticleDefinition* particle = particleTable->FindParticle(particleName =
    ↪  "gamma");
    fParticleGun->SetParticleDefinition(particle);

    // Set the particle position
    G4double posZ = -80.5 * cm;
    fParticleGun->SetParticlePosition(G4ThreeVector(0, 0, posZ));
}

ICRP110PhantomPrimaryGeneratorAction::~ICRP110PhantomPrimaryGeneratorAction() {
    delete fParticleGun;
}

void ICRP110PhantomPrimaryGeneratorAction::GeneratePrimaries(G4Event* anEvent) {
    // Generate energy uniformly distributed between 10 and 140 keV
    G4double minEnergy = 10. * keV;
    G4double maxEnergy = 140. * keV;
    G4double energy = minEnergy + G4UniformRand() * (maxEnergy - minEnergy);
    fParticleGun->SetParticleEnergy(energy);

    // Generate direction uniformly distributed in a cone of 16 degrees around
    ↪  the z-axis
    G4double coneAngle = 16. * degree;
    G4double pi = 3.141592654;
    G4double phi = G4UniformRand() * 2. * pi;
    G4double theta = acos(1. - G4UniformRand() * (1. - cos(coneAngle)));
```

```
    G4ThreeVector direction(sin(theta) * cos(phi), sin(theta) * sin(phi),
    ↪  cos(theta));
    fParticleGun->SetParticleMomentumDirection(direction);
}
```

A.2.6 Physics list

The constructor of the SimplePhysicsList class calls the constructor of G4VModularPhysicsList. With the Construct Particle method, it defines the particles of interest for the simulation: G4Gamma photons, electrons and positrons; it constructs the physics processes: it adds particle transport, for gotons it constructs two processes: CustomPhotoElectricEffect for the photoelectric effect and CustomComptonScattering for Compton scattering; for electrons G4eMultipleScattering and G4eIonisation; also sets cut-off values to 0.3mm

```
#include "SimplePhysicsList.hh"
#include "CustomComptonScattering.hh"
#include "CustomPhotoElectricEffect.hh"
#include "G4ProcessManager.hh"
#include "G4SystemOfUnits.hh"
#include "G4Gamma.hh"
#include "G4Electron.hh"
#include "G4Positron.hh"

// Includi i file header per G4eMultipleScattering e G4eIonisation
#include "G4eMultipleScattering.hh"
#include "G4eIonisation.hh"

SimplePhysicsList::SimplePhysicsList() : G4VModularPhysicsList() {
    SetVerboseLevel(1);
}

SimplePhysicsList::~SimplePhysicsList() {}

void SimplePhysicsList::ConstructParticle() {
    G4Gamma::GammaDefinition();
    G4Electron::ElectronDefinition();
    G4Positron::PositronDefinition();
}

void SimplePhysicsList::ConstructProcess() {
    AddTransportation();

    G4ProcessManager *phManager =
    ↪  G4Gamma::GammaDefinition()->GetProcessManager();

    // Utilizzo delle classi personalizzate
    phManager->AddDiscreteProcess(new CustomPhotoElectricEffect);
    phManager->AddDiscreteProcess(new CustomComptonScattering);

    G4ProcessManager *elManager =
    ↪  G4Electron::ElectronDefinition()->GetProcessManager();
    elManager->AddProcess(new G4eMultipleScattering, -1, 1, 1);
    elManager->AddProcess(new G4eIonisation, -1, 2, 2);
}
```

```
void SimplePhysicsList::SetCuts() {
    defaultCutValue = 0.3 * mm;
    SetCutsWithDefault();
}
```

A.2.7 Custom PhotoElectric Effect

The constructor of the CustomPhotoElectricEffect class calls the constructor of the G4PhotoElectricEffect. calls the PostStepDoIt method of the base class G4PhotoElectricEffect, which is called after a step (step) of the simulation has been completed, which handles the normal behaviour of the photoelectric effect, such as the production of the ejected electron and the handling of the initial photon. However, after the call to the basic method it forces the primary particle (the photon) to stop (fStopAndKill). This means that the trace of the photon is interrupted and is not simulated any further. The code runs through all secondary particles generated by the process (such as the ejected electron) and kills them too, setting their state to fStopAndKill. This ensures that no secondary particles are tracked beyond the initial interaction.

```
#include "CustomPhotoElectricEffect.hh"
#include "G4Track.hh"
#include "G4Step.hh"
#include "G4ParticleChange.hh"

CustomPhotoElectricEffect::CustomPhotoElectricEffect() : G4PhotoElectricEffect()
↪  {}

CustomPhotoElectricEffect::~CustomPhotoElectricEffect() {}

G4VParticleChange* CustomPhotoElectricEffect::PostStepDoIt(const G4Track&
↪  aTrack, const G4Step& aStep) {
    G4VParticleChange* particleChange =
    ↪  G4PhotoElectricEffect::PostStepDoIt(aTrack, aStep);

    // Uccidi la particella primaria
    particleChange->ProposeTrackStatus(fStopAndKill);

    // Uccidi eventuali particelle secondarie
    for (size_t i = 0; i < particleChange->GetNumberOfSecondaries(); ++i) {
        particleChange->GetSecondary(i)->SetTrackStatus(fStopAndKill);
    }

    return particleChange;
}
```

A.2.8 Custom Compton Scattering

The constructor of the CustomComptonScattering class calls the constructor of G4ComptonScattering. calls the PostStepDoIt method of the base class G4PhotoElectricEffect, which is called after a step (step) of the simulation has been completed, which handles the normal behaviour of Compton scattering, such as calculating the new energy and direction of the photon, and the generation of any secondary particles (e.g. the electron). However, after the call to the basic method it forces the primary particle (the photon) to stop (fStopAndKill). This means that the trace of the photon is interrupted and is not

simulated any further. The code runs through all secondary particles generated by the process (such as the electron produced by scattering) and kills them too, setting their state to fStopAndKill. This ensures that no secondary particles are tracked beyond the initial interaction.

```
#include "CustomComptonScattering.hh"
#include "G4Track.hh"
#include "G4Step.hh"
#include "G4ParticleChange.hh"

CustomComptonScattering::CustomComptonScattering() : G4ComptonScattering() {}

CustomComptonScattering::~CustomComptonScattering() {}

G4VParticleChange* CustomComptonScattering::PostStepDoIt(const G4Track& aTrack,
↪  const G4Step& aStep) {
    G4VParticleChange* particleChange =
    ↪  G4ComptonScattering::PostStepDoIt(aTrack, aStep);

    // Uccidi la particella primaria
    particleChange->ProposeTrackStatus(fStopAndKill);

    // Uccidi eventuali particelle secondarie
    for (size_t i = 0; i < particleChange->GetNumberOfSecondaries(); ++i) {
        particleChange->GetSecondary(i)->SetTrackStatus(fStopAndKill);
    }

    return particleChange;
}
```

Appendix B

Volume reconstruction scripts

In this appendix, will be discussed the codes, that can also be found in the GitHub repository [96], used to go from the 360 projections, one for each degree of phantom rotation obtained with the MC methods, to the files OSCaR needs to perform the volumetric reconstruction.

B.1 Gate

The code below, used with GATE converts text output files to grayscale images:

- **Reads text files**: Each line is parsed into pixel values.
- **Creates a 200x200 grayscale image**: Pixel values from the text file are used to set pixel intensity.
- **Saves the image as JPEG**: The image is saved with the same name as the text file.
- **Processes all text files in a folder**: Converts each `.txt` file in the specified folder to a `.jpg` image.

```
import os
from PIL import Image

def text_to_image(input_file, output_file):
    # Leggi il file di testo
    with open(input_file, 'r') as file:
        lines = file.readlines()

    # Crea un'immagine vuota 200x200 pixel in scala di grigi
    image = Image.new('L', (200, 200))

    # Converti i numeri nel file di testo in pixel dell'immagine
    for y, line in enumerate(lines):
```

```
        values = line.strip().split()[:200]  # Considera solo i primi 200 numeri
        ↪ per riga
        for x, value in enumerate(values):
                image.putpixel((x, y), int(value))

    # Salva l'immagine come file JPEG
    image.save(output_file)

# Cartella di input
input_folder = r'E:/marco/unipv/tesi/output'

# Ciclo su tutti i file .txt nella cartella di input
for filename in os.listdir(input_folder):
    if filename.endswith('.txt'):
        input_file = os.path.join(input_folder, filename)
        output_file = os.path.join(input_folder,
        ↪ f"{os.path.splitext(filename)[0]}.jpg")
        text_to_image(input_file, output_file)
```

From all the images can be constructed the .csv file with the code [110]:

```
import os
import csv
from PIL import Image
import re

def leggi_massimo_pixel(cartella):
    massimo_pixel = 0
    for filename in os.listdir(cartella):
        if filename.endswith(".jpg"):
            path_file = os.path.join(cartella, filename)
            try:
                with Image.open(path_file) as img:
                    extrema = img.getextrema()
                    if isinstance(extrema, tuple) and len(extrema) == 2:
                        # Per immagini in scala di grigi, extrema sarà (min,
                        ↪  max)    88 e 97
                        massimo_pixel = max(massimo_pixel, extrema[1])
                    else:
                        # Per immagini a colori o altri formati
                        massimo_pixel = max(massimo_pixel, max(max(e) for e in
                        ↪  extrema))
            except Exception as e:
                print(f"Errore nell'aprire l'immagine {filename}: {e}")
    return massimo_pixel

def crea_csv_da_cartella(cartella):
    if not os.path.exists(cartella):
        print(f"La cartella {cartella} non esiste.")
        return

    file_list = []
    for filename in os.listdir(cartella):
        if filename.endswith(".jpg"):
            file_list.append(filename)

    # Ordinare la lista dei file usando il primo numero trovato nel nome del
    ↪  file
    file_list.sort(key=lambda x: float(re.search(r'\d+\.*\d*', x).group()) if
    ↪  re.search(r'\d+\.*\d*', x) else 0)

    massimo_pixel = leggi_massimo_pixel(cartella)

    output_file_path = os.path.join(cartella, 'output.csv')

    try:
        with open(output_file_path, mode='w', newline='') as file:
            writer = csv.writer(file, delimiter=',', lineterminator='\n',
            ↪  quoting=csv.QUOTE_MINIMAL, quotechar='"', escapechar='\\',
            ↪  doublequote=True, skipinitialspace=False)
```

```
            for filename in file_list:
                nome_file = filename
                angolo = float(re.search(r'\d+\.*\d*', filename).group()) if
                ↪  re.search(r'\d+\.*\d*', filename) else 0
                valore1 = '65'
                valore2 = '65'
                writer.writerow([nome_file, angolo, valore1, valore2,
                ↪  massimo_pixel, '1'])
    except Exception as e:
        print(f"Errore nella scrittura del file CSV: {e}")
    print("Done")

crea_csv_da_cartella('')
```

B.2 Geant4

The code below, used with Geant4 converts ROOT histogram files to grayscale images:

- **Reads ROOT files**: Opens and extracts 2D histogram data.
- **Normalizes histogram data**: Scales values to the range $[0, 255]$.
- **Creates and saves grayscale images**: Converts normalized data to images and saves as JPEG files.
- **Processes all ROOT files in a folder**: Extracts and converts histograms from each `.root` file in the specified input folder.

```
import os
import re
import uproot
import numpy as np
from PIL import Image

def root_hist_to_image(input_file, output_file, hist_name):
    # Apri il file ROOT e accedi all'istogramma 2D
    with uproot.open(input_file) as file:
        hist = file[hist_name]

        # Ottieni i dati dell'istogramma come un array NumPy
        values = hist.values()

        values = (values / np.max(values)) * 255  # Scala tra 0 e 255

        # Crea un'immagine dall'array normalizzato
        image = Image.fromarray(values.astype(np.uint8), mode='L')

        # Salva l'immagine come file JPEG
        image.save(output_file)

# Cartella di input (modifica questa variabile con il percorso della tua
↪  cartella contenente i file ROOT)
input_folder = r''

# Cartella di output
output_folder = r''

# Nome dell'istogramma da estrarre
histogram_name = 'PhotonPosition'  # Nome dell'istogramma 2D nel file ROOT

# Controlla se la cartella di output esiste
if not os.path.exists(output_folder):
    print(f"Errore: La cartella di output '{output_folder}' non esiste.")
else:
    # Ciclo su tutti i file .root nella cartella di input
    for filename in os.listdir(input_folder):
```

```
        if filename.endswith('.root'):
            # Estrai il numero 'n' dal nome del file (es. output_n.root)
            match = re.match(r'output_(\d+).root', filename)
            if match:
                n = match.group(1)  # Ottiene il numero "n" dal nome del file
                input_file = os.path.join(input_folder, filename)
                output_file = os.path.join(output_folder, f"{n}.jpg")
                root_hist_to_image(input_file, output_file, histogram_name)

    print("Conversione completata!")
```

The following code automates the rotation of ICRP110 on Geant4 by modifying the line containing 'rotCylinder->rotateZ(' in the file ICRP110PhantomConstruction.cc and modifying the name of the output ROOT file in the file ICRP110PhantomAnalysisManager.cc. Using the 'subprocess' module with the function subprocess.run(['make'], check=True), it executes the make command and raises an exception if the compilation fails; after compiling with subprocess.run it runs the simulation './ICRP110phantoms :

```
import subprocess

# Percorso del file da modificare
file_path = '../src/ICRP110PhantomConstruction.cc'
analysis_path = '../src/ICRP110PhantomAnalysisManager.cc'

#for angle in range(4):
for angle in [0, 45, 90, 180, 270]:
    # Legge il contenuto del file ICRP110PhantomConstruction.cc
    with open(file_path, 'r') as file:
        lines = file.readlines()

    # Modifica la linea desiderata
    for i, line in enumerate(lines):
        if "rotCylinder->rotateZ(" in line:
            lines[i] = f'rotCylinder->rotateZ({angle} * deg); // Modified
            ↪  rotation\n'
            break

    # Sovrascrive il file con la nuova rotazione
    with open(file_path, 'w') as file:
        file.writelines(lines)

    # Legge il file di analisi
    with open(analysis_path, 'r') as file:
        lines = file.readlines()

    # Modifica il file di analisi per cambiare il nome del file output.root
    updated = False
    for i, line in enumerate(lines):
        if 'new TFile(' in line and 'output_' in line and '.root' in line:
            lines[i] = f'TFile* file = new TFile("output_{angle}.root",
            ↪  "RECREATE");\n'
```

```
            updated = True
            break

    # Se non viene trovata la linea da aggiornare, cerca la stringa originale
    if not updated:
        for i, line in enumerate(lines):
            if 'new TFile("output.root"' in line:
                lines[i] = f'TFile* file = new TFile("output_{angle}.root",
                ↪  "RECREATE");\n'
                break

    # Sovrascrive il file di analisi con la nuova riga
    with open(analysis_path, 'w') as file:
        file.writelines(lines)

    # Compila il codice modificato con make
    print(f"Compiling for rotation {angle} degrees...")
    subprocess.run(["make"], check=True)

    # Esegue la simulazione con il file macro
    print(f"Running simulation for rotation {angle} degrees...")
    subprocess.run(["./ICRP110phantoms", "run.mac"], check=True)
```

Bibliography

[1] Anne Aupérin. "Epidemiology of head and neck cancers: an update". In: *Current opinion in oncology* 32.3 (2020), pp. 178–186.

[2] Nadarajah Vigneswaran and Michelle D Williams. "Epidemiologic trends in head and neck cancer and aids in diagnosis". In: *Oral and Maxillofacial Surgery Clinics* 26.2 (2014), pp. 123–141.

[3] Laura QM Chow. "Head and neck cancer". In: *New England Journal of Medicine* 382.1 (2020), pp. 60–72.

[4] Itsuro Kato et al. "Effectiveness of BNCT for recurrent head and neck malignancies". In: *Applied Radiation and Isotopes* 61.5 (2004), pp. 1069–1073.

[5] IAEA. *Advances in Boron Neutron Capture Therapy*. IAEA., June 2023. ISBN: 978-92-0-132723-9.

[6] Dmitry Kasatov et al. "The accelerator neutron source for boron neutron capture therapy". In: *Journal of Physics: Conference Series* 769 (Nov. 2016), p. 012064. DOI: 10.1088/1742-6596/769/1/012064.

[7] G Locher. "Ab-x nct". In: *Am. J. Roentgenol* 36 (1936), pp. 1–13.

[8] Jeffrey A Coderre et al. "Boron neutron capture therapy: cellular targeting of high linear energy transfer radiation". en. In: *Technol. Cancer Res. Treat.* 2.5 (Oct. 2003), pp. 355–375.

[9] Minoru Suzuki. "Boron neutron capture therapy (BNCT): A unique role in radiotherapy with a view to entering the accelerator-based BNCT era". In: *International journal of clinical oncology* 25.1 (2020), pp. 43–50.

[10] AH Soloway, H Hatanaka, and MA Davis. "Penetration of brain and brain tumor. VII. Tumor-binding sulfhydryl boron compounds". In: *Journal of medicinal chemistry* 10.4 (1967), pp. 714–717.

[11] Hiroshi Hatanaka. "A revised boron-neutron capture therapy for malignant brain tumors: II. Interim clinical result with the patients excluding previous treatments". In: *Journal of neurology* 209 (1975), pp. 81–94.

[12] Yutaka Mishima et al. "Treatment of malignant melanoma by single thermal neutron capture therapy with melanoma-seeking 10B-compound". In: *The Lancet* 334.8659 (1989), pp. 388–389.

[13] Shinji Kawabata et al. "Boron Compounds for Neutron Capture Therapy in the Treatment of Brain Tumors". In: Aug. 2022. ISBN: 978-1-80356-464-7. DOI: 10.5772/intechopen.106202.

[14] Hiroaki Kumada and Kenta Takada. "Treatment planning system and patient positioning for boron neutron capture therapy". In: *Therapeutic Radiology and Oncology* 2 (2018).

[15] Ian Postuma et al. "Using the photon isoeffective dose formalism to compare and combine BNCT and CIRT in a head and neck tumour". In: *Scientific Reports* 14.1 (2024), p. 418.

[16] Zizhu Zhang and Tong Liu. "A review of the development of In-Hospital Neutron Irradiator-1 and boron neutron capture therapy clinical research on malignant melanoma". In: *Therapeutic Radiology and Oncology* 2 (2018).

[17] Rolf F Barth et al. "Boron neutron capture therapy of cancer: current status and future prospects". In: *Clinical Cancer Research* 11.11 (2005), pp. 3987–4002.

[18] Cancer Council Australia. *Anatomy of the head and neck*. URL: `https://www.cancercouncil.com.au/head-and-neck-cancer/about-head-and-neck-cancers/`.

[19] Sara Gandini et al. "Tobacco smoking and cancer: a meta-analysis". In: *International journal of cancer* 122.1 (2008), pp. 155–164.

[20] Loredana G Marcu et al. *Radiotherapy and clinical radiobiology of head and neck cancer*. CRC Press, 2018.

[21] Anil K Chaturvedi et al. "Human papillomavirus and rising oropharyngeal cancer incidence in the United States". In: *Journal of clinical oncology* 29.32 (2011), pp. 4294–4301.

[22] Per Attner et al. "The role of human papillomavirus in the increased incidence of base of tongue cancer". In: *International journal of cancer* 126.12 (2010), pp. 2879–2884.

[23] Neela Guha et al. "Oral health and risk of squamous cell carcinoma of the head and neck and esophagus: results of two multicentric case-control studies". In: *American journal of epidemiology* 166.10 (2007), pp. 1159–1173.

[24] HR Snyder, Albert J Reedy, and Wm J Lennarz. "Synthesis of aromatic boronic acids. aldehydo boronic acids and a boronic acid analog of tyrosine1". In: *Journal of the American Chemical Society* 80.4 (1958), pp. 835–838.

[25] D.R. Dance and International Atomic Energy Agency. *Diagnostic Radiology Physics: A Handbook for Teachers and Students*. STI/PUB. International Atomic Energy Agency, 2014. ISBN: 9789201310101. URL: `https://books.google.it/books?id=9iiKmwEACAAJ`.

[26] H Tanaka et al. "Characteristics comparison between a cyclotron-based neutron source and KUR-HWNIF for boron neutron capture therapy". In: *Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms* 267.11 (2009), pp. 1970–1977.

[27] Minoru Suzuki et al. "Impact of intra-arterial administration of boron compounds on dose–volume histograms in boron neutron capture therapy for recurrent head-and-neck tumors". In: *International Journal of Radiation Oncology* Biology* Physics* 66.5 (2006), pp. 1523–1527.

[28] Leena Kankaanranta et al. "Boron neutron capture therapy in the treatment of locally recurred head-and-neck cancer: final analysis of a phase I/II trial". In:

International Journal of Radiation Oncology Biology* Physics* 82.1 (2012), e67–e75.

[29] LW Wang et al. "BNCT for locally recurrent head and neck cancer: preliminary clinical experience from a phase I/II trial at Tsing Hua open-pool reactor". In: *Applied Radiation and Isotopes* 69.12 (2011), pp. 1803–1806.

[30] Ling-Wei Wang et al. "Fractionated BNCT for locally recurrent head and neck cancer: Experience from a phase I/II clinical trial at Tsing Hua Open-Pool Reactor". In: *Applied Radiation and Isotopes* 88 (2014), pp. 23–27.

[31] Ling-Wei Wang et al. "Clinical trials for treating recurrent head and neck cancer with boron neutron capture therapy using the Tsing-Hua Open Pool Reactor". In: *Cancer Communications* 38 (2018), pp. 1–7.

[32] Katsumi Hirose et al. "Boron neutron capture therapy using cyclotron-based epithermal neutron source and borofalan (10B) for recurrent or locally advanced head and neck cancer (JHN002): An open-label phase II trial". In: *Radiotherapy and Oncology* 155 (2021), pp. 182–187.

[33] S Vatnitsky. *Specification and acceptance testing of radiotherapy treatment planning systems*. IAEA, 2007.

[34] Ervin B Podgorsak. *Radiation oncology physics: a handbook for teachers and students*. 2005.

[35] Philip Mayles, Alan Nahum, and Jean-Claude Rosenwald. *Handbook of radiotherapy physics: theory and practice*. CRC Press, 2007.

[36] Andrée Dutreix. "When and how can we improve precision in radiotherapy?" In: *Radiotherapy and Oncology* 2.4 (1984), pp. 275–292. ISSN: 0167-8140. DOI: https://doi.org/10.1016/S0167-8140(84)80070-5. URL: https://www.sciencedirect.com/science/article/pii/S0167814084800705.

[37] C. Guy and D. Ffytche. *An Introduction to the Principles of Medical Imaging*. Imperial College Press, 2005. ISBN: 9781860945021. URL: https://books.google.it/books?id=-jt1QgAACAAJ.

[38] Rita R. "X-ray imaging using 100 um thick Gas Electron Multipliers operating in Kr-CO2 mixtures". PhD thesis. June 2018. DOI: 10.13140/RG.2.2.16794.49600.

[39] P.J. Allisy-Roberts and J. Williams. *Farr's Physics for Medical Imaging*. Elsevier Health Sciences, 2007. ISBN: 9780702028441. URL: https://books.google.it/books?id=EHODwuD73XMC.

[40] Thomas R Fewell. *Handbook of computed tomography x-ray spectra*. US Department of Health and Human Services, Public Health Service, Food and ..., 1981.

[41] Prashant Kore, Pravina Pawar, and T. Selvam. "Evaluation of radiological data of some saturated fatty acids using gamma ray spectrometry". In: *Radiation Physics and Chemistry* 119 (Feb. 2016), pp. 74–79. DOI: 10.1016/j.radphyschem.2015.09.003.

[42] Hiwa Qadr. "Calculation for gamma ray buildup factor for aluminium, graphite and lead". In: *International Journal of Nuclear Energy Science and Technology* 13 (Apr. 2019), p. 61. DOI: 10.1504/IJNEST.2019.099718.

[43] Sharaf J.M. and Saleh H. "Gamma-ray energy buildup factor calculations and shielding effects of some Jordanian building structures". In: *Radiation Physics*

and Chemistry 110 (2015), pp. 87–95. ISSN: 0969-806X. DOI: https://doi.org/10.1016/j.radphyschem.2015.01.031. URL: https://www.sciencedirect.com/science/article/pii/S0969806X15000389.

[44] M.A. Flower. *Webb's Physics of Medical Imaging, Second Edition*. Series in Medical Physics and Biomedical Engineering. Taylor & Francis, 2012. ISBN: 9780750305730. URL: https://books.google.it/books?id=qkFljemf7y0C.

[45] W.R. Hendee and E.R. Ritenour. *Medical Imaging Physics*. Wiley, 2003. ISBN: 9780471461135. URL: https://books.google.it/books?id=5Twgg8NBtrMC.

[46] William R Hendee and Raymond P Rossi. *Quality assurance for conventional tomographic x-ray units*. Department of Health, Education, and Welfare, Public Health Service, Food an ..., 1979.

[47] A. Rose. *Vision: Human and Electronic*. Ballard CREOL collection. Springer US, 1974. ISBN: 9780306307324. URL: https://books.google.it/books?id=s0GjM_rY95kC.

[48] R.J. Sweeney. *Radiographic Artifacts: Their Cause and Control*. Lippincott, 1983. ISBN: 9780397505548. URL: https://books.google.it/books?id=iYVrAAAAMAAJ.

[49] C.L. Morgan and M.D. Miller. *Basic Principles of Computed Tomography*. University Park Press, 1983. ISBN: 9780839117056. URL: https://books.google.it/books?id=fhLwpwAACAAJ.

[50] Hounsfield Godfrey N. "Computed Medical Imaging". In: *Science* 210.4465 (1980), pp. 22–28. DOI: 10.1126/science.6997993. eprint: https://www.science.org/doi/pdf/10.1126/science.6997993. URL: https://www.science.org/doi/abs/10.1126/science.6997993.

[51] C J Martin. "Computed tomography, 2nd edn. By WA Kalender. pp.306, 2005 (Publicis, Erlangen, Germany) €49.90 ISBN 3-89578-216-5". In: *British Journal of Radiology* 79.945 (Jan. 2014), pp. 774–774. ISSN: 0007-1285. DOI: 10.1259/bjr.79.945.774. URL: https://academic.oup.com/bjr/article-pdf/79/945/774/54469848/bjr.79.945.774.pdf.

[52] Thorsten Buzug and Dimitris Mihailidis. "Computed Tomography From Photon Statistics to Modern Cone-Beam CT". In: *Medical Physics* 36 (Aug. 2009), pp. 3858–. DOI: 10.1118/1.3176026.

[53] Seeram E. *Computed Tomography - E-Book: Computed Tomography - E-Book*. Elsevier Health Sciences, 2015. ISBN: 9780323323017. URL: https://books.google.it/books?id=DTCDCgAAQBAJ.

[54] Nooshin Kiarashi and Ehsan Samei. "Digital breast tomosynthesis: A concise overview". In: *Imaging in Medicine* 5 (Oct. 2013), pp. 467–476. DOI: 10.2217/iim.13.52.

[55] CEFLA S.C. NEWTOM. *NEWTOM PRESENTS THE FIRST MULTI- SCAN BODY CBCT FOR EXTENSIVE CLINICAL APPLICATIONS*. 2022. URL: https://www.newtom.it/en/medicale/prodotti/7g/.

[56] Amadeus Alcantara et al. "Patient-Specific Bone Multiscale Modelling, Fracture Simulation and Risk Analysis—A Survey". In: *Materials* 13 (Dec. 2019), p. 106. DOI: 10.3390/ma13010106.

[57] Gwen RJ Swennen and Filip Schutyser. "Three-dimensional cephalometry: spiral multi-slice vs cone-beam computed tomography". In: *American Journal of Orthodontics and Dentofacial Orthopedics* 130.3 (2006), pp. 410–416.

[58] W De Vos, Jan Casselman, and GRJ19464146 Swennen. "Cone-beam computerized tomography (CBCT) imaging of the oral and maxillofacial region: a systematic review of the literature". In: *International journal of oral and maxillofacial surgery* 38.6 (2009), pp. 609–625.

[59] G.T. Herman. *Fundamentals of Computerized Tomography: Image Reconstruction from Projections*. Advances in Computer Vision and Pattern Recognition. Springer London, 2009. ISBN: 9781846287237. URL: https://books.google.it/books?id=BhtGTkEjkOQC.

[60] W.R. Hendee. *The Physical Principles of Computed Tomography*. Little, Brown library of radiology. Little, Brown, 1983. ISBN: 9780316355940. URL: https://books.google.it/books?id=ijcQAQAAMAAJ.

[61] Samuel Matej, Jeffrey A Fessler, and Ivan G Kazantsev. "Iterative tomographic image reconstruction using Fourier-based forward and back-projectors". In: *IEEE transactions on medical imaging* 23.4 (2004), pp. 401–412.

[62] Mita D Desai and W Kenneth Jenkins. "Convolution backprojection image reconstruction for spotlight mode synthetic aperture radar". In: *IEEE Transactions on Image Processing* 1.4 (1992), pp. 505–517.

[63] C.C. Shaw. *Cone Beam Computed Tomography*. Imaging in medical diagnosis and therapy. Taylor & Francis, 2014. ISBN: 9781439846261. URL: https://books.google.it/books?id=pqSNAgAAQBAJ.

[64] Lee Feldkamp, L. C. Davis, and James Kress. "Practical Cone-Beam Algorithm". In: *J. Opt. Soc. Am* 1 (Jan. 1984), pp. 612–619.

[65] Gengsheng Lawrence Zeng and Gengsheng Lawrence Zeng. "3D image reconstruction". In: *Medical Image Reconstruction: A Conceptual Tutorial* (2010), pp. 87–123.

[66] Gengsheng Zeng, Yi Weng, and Grant Gullberg. "Iterative reconstruction with attenuation compensation from cone-beam projections acquired via nonplanar orbits". In: *Nuclear Science, IEEE Transactions on* 44 (Mar. 1997), pp. 98–106. DOI: 10.1109/23.554832.

[67] N Rezvani et al. "OSCaR: An open-source cone-beam CT reconstruction tool for imaging research". In: *Medical physics* 34.6Part2 (2007), pp. 2341–2341.

[68] DICOM digital imaging and communications in medicine. *About DICOM: Overview*. https://www.dicomstandard.org/about-home.

[69] Dorin Andrica et al. "More on Second-Order Linear Recurrent Sequences". In: *Recurrent Sequences: Key Results, Applications, and Problems* (2020), pp. 135–194.

[70] N Metropolis. "The beginning". In: *Los Alamos Science* 15 (1987), pp. 125–130.

[71] Johannes Bentner et al. "Optimization of the time-dependent traveling salesman problem with Monte Carlo methods". In: *Physical Review E* 64.3 (2001), p. 036701.

[72] Necia Grant Cooper, Roger Eckhardt, and Nancy Shera. *From cardinals to chaos: reflection on the life and legacy of Stanislaw Ulam*. CUP Archive, 1989.

[73] Alberto Rotondi, Paolo Pedroni, and Antonio Pievatolo. *Probability, Statistics and Simulation: With Application Programs Written in R*. Vol. 139. Springer Nature, 2022.

[74] Nicholas Metropolis et al. "The beginning of the Monte Carlo method". In: *Los Alamos Science* 15.584 (1987), pp. 125–130.

[75] Sébastien Jan et al. "GATE V6: a major enhancement of the GATE simulation platform enabling modelling of CT and radiotherapy". In: *Physics in Medicine & Biology* 56.4 (2011), p. 881.

[76] Sea Agostinelli et al. "GEANT4—a simulation toolkit". In: *Nuclear instruments and methods in physics research section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 506.3 (2003), pp. 250–303.

[77] John Allison et al. "Geant4 developments and applications". In: *IEEE Transactions on nuclear science* 53.1 (2006), pp. 270–278.

[78] John Apostolakis et al. "Geometry and physics of the geant4 toolkit for high and medium energy applications". In: *Radiation Physics and Chemistry* 78.10 (2009), pp. 859–873.

[79] Antony J Lomax et al. "A treatment planning inter-comparison of proton and intensity modulated photon radiotherapy". In: *Radiotherapy and oncology* 51.3 (1999), pp. 257–271.

[80] JP Wellisch. "Geant4 physics validation for large HEP detectors". In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 502.2-3 (2003), pp. 669–672.

[81] Till Tobias Böhlen et al. "Benchmarking nuclear models of FLUKA and GEANT4 for carbon ion therapy". In: *Physics in Medicine & Biology* 55.19 (2010), p. 5833.

[82] Clermont-Fd France Lydia Maigne University Clermont Auvergne. *The OpenGATE collaboration*. URL: `http://www.opengatecollaboration.org/`.

[83] Eugen B Hug. "Protons versus photons: a status assessment at the beginning of the 21st Century." In: *Radiotherapy and Oncology: Journal of the European Society for Therapeutic Radiology and Oncology* 73 (2004), S35–7.

[84] David Sarrut et al. "The OpenGATE ecosystem for Monte Carlo simulation in medical physics". en. In: *Phys. Med. Biol.* 67.18 (Sept. 2022), p. 184001.

[85] Matthias Fippel and Martin Soukup. "A Monte Carlo dose calculation algorithm for proton therapy". In: *Medical Physics* 31.8 (2004), pp. 2263–2273. DOI: `https://doi.org/10.1118/1.1769631`. eprint: `https://aapm.onlinelibrary.wiley.com/doi/pdf/10.1118/1.1769631`. URL: `https://aapm.onlinelibrary.wiley.com/doi/abs/10.1118/1.1769631`.

[86] J. Allison et al. "Recent developments in Geant4". In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 835 (2016), pp. 186–225. ISSN: 0168-9002. DOI: `https://doi.org/10.1016/j.nima.2016.06.125`. URL: `https://www.sciencedirect.com/science/article/pii/S0168900216306957`.

[87] Hans-Georg Menzel. "ICRP Publication 110. Realistic reference phantoms: an ICRP/ICRU joint effort. A report of adult reference computational phantoms.

Ann". In: *ICRP* 39 (2009), p. 1. URL: https://www.icrp.org/publication.asp?id=ICRP%20Publication%20110.

[88] Matthew Large et al. "Modelling ICRP110 Adult Reference Voxel Phantoms for dosimetric applications: Development of a new Geant4 Advanced Example". In: vol. 1662. Oct. 2020. DOI: 10.1088/1742-6596/1662/1/012021.

[89] David Guez et al. "Counting rates modeling for PET scanners with GATE". In: *Nuclear Science, IEEE Transactions on* 55 (Mar. 2008), pp. 516–523. DOI: 10.1109/TNS.2007.910880.

[90] Sébastien Jan et al. "GATE: a simulation toolkit for PET and SPECT". In: *Physics in Medicine & Biology* 49.19 (2004), p. 4543.

[91] Steven Staelens and Irène Buvat. "Chapter 5 - Monte Carlo Simulations in Nuclear Medicine Imaging". In: *Advances in Biomedical Engineering*. Ed. by Pascal Verdonck. Amsterdam: Elsevier, 2009, pp. 177–209. ISBN: 978-0-444-53075-2. DOI: https://doi.org/10.1016/B978-0-444-53075-2.00005-8. URL: https://www.sciencedirect.com/science/article/pii/B9780444530752000058.

[92] Olga Kochebina et al. "New GATE Digitizer Unit for versions post v9. 3". In: *Frontiers in Physics* 12 (2024), p. 1294916.

[93] GateContrib. *GateContrib: Imaging/CT/Classic*. https://github.com/OpenGATE/GateContrib/tree/045f7663abd50191383220fd3ab9f0a801d2ee0e/imaging/CT/classic. 2024.

[94] Bangho Shin et al. "New skeletal dose coefficients of the ICRP-110 reference phantoms for idealized external fields to photons and neutrons using dose response functions (DRFs)". In: *Nuclear Engineering and Technology* 55.6 (2023), pp. 1949–1958. ISSN: 1738-5733. DOI: https://doi.org/10.1016/j.net.2023.03.015. URL: https://www.sciencedirect.com/science/article/pii/S1738573323001249.

[95] Susanna Guatelli et al. "Introduction to the Geant4 Simulation toolkit". In: *AIP Conference Proceedings*. Vol. 1345. 1. American Institute of Physics. 2011, pp. 303–322.

[96] Paganelli Marco. *ICRP110CBCT*. https://github.com/paganellimarco/ICRP110_CBCT.git. 2024.

[97] Paganelli Marco. *Multi-thread Issue in ICRP110 HumanPhantom*. https://geant4-forum.web.cern.ch/t/multi-thread-issue-in-icrp110-humanphantom/11207. 2023.

[98] Steve Pieper, M. Halle, and Ron Kikinis. "3D slicer". In: vol. vol. 1. May 2004, 632–635 Vol. 1. DOI: 10.1109/ISBI.2004.1398617.

[99] Andreas Kaplan and Michael Haenlein. "Siri, Siri, in my hand: Who's the fairest in the land? On the interpretations, illustrations, and implications of artificial intelligence". In: *Business horizons* 62.1 (2019), pp. 15–25.

[100] Tackeun Kim et al. "Machine learning for detecting moyamoya disease in plain skull radiography using a convolutional neural network". In: *EBioMedicine* 40 (2019), pp. 636–642. ISSN: 2352-3964. DOI: https://doi.org/10.1016/j.ebiom.2018.12.043. URL: https://www.sciencedirect.com/science/article/pii/S2352396418306200.

[101] Wenya Linda Bi et al. “Artificial intelligence in cancer imaging: clinical challenges and applications”. In: *CA: a cancer journal for clinicians* 69.2 (2019), pp. 127–157.

[102] Awais Mansoor et al. “Segmentation and image analysis of abnormal lungs at CT: current approaches, challenges, and future trends”. In: *Radiographics* 35.4 (2015), pp. 1056–1076.

[103] Hui Lin et al. “Deep learning for automatic target volume segmentation in radiation therapy: a review”. In: *Quantitative Imaging in Medicine and Surgery* 11.12 (2021), p. 4847.

[104] K Harrison et al. “Machine learning for auto-segmentation in radiotherapy planning”. In: *Clinical Oncology* 34.2 (2022), pp. 74–88.

[105] Yabo Fu et al. “A review of deep learning based methods for medical image multi-organ segmentation”. In: *Physica Medica* 85 (2021), pp. 107–122.

[106] Ozan Oktay et al. “Evaluation of deep learning to augment image-guided radiotherapy for head and neck and prostate cancers”. In: *JAMA network open* 3.11 (2020), e2027426–e2027426.

[107] Kenneth Clark et al. “The Cancer Imaging Archive (TCIA): maintaining and operating a public information repository”. In: *Journal of digital imaging* 26 (2013), pp. 1045–1057.

[108] Chan Hyeong Kim et al. “ICRP publication 145: adult mesh-type reference computational phantoms”. In: *Annals of the ICRP* 49.3 (2020), pp. 13–201.

[109] Paganelli Marco. *ROOT TO TXT*. `https://github.com/paganellimarco/ICRP110_CBCT/blob/526d650e404607f6db7350b16d4c8db87e0f6b83/Analyze.cpp`. 2024.

[110] Paganelli Marco. *create CSV*. `https://github.com/paganellimarco/ICRP110_CBCT/blob/526d650e404607f6db7350b16d4c8db87e0f6b83/csv_da_cartella.py`. 2024.

Ringraziamenti

Alla conclusione di questo elaborato, desidero esprimere la mia più sincera gratitudine a tutte le persone senza le quali questo lavoro di tesi non avrebbe mai potuto vedere la luce.
In primo luogo, il mio più profondo ringraziamento va a Silva, Ian e Seta per aver creduto nella mia idea e per avermi sostenuto fin dal primo giorno, accompagnandomi fino all'ultimo istante. Un sentito grazie anche a Daniele, che, nonostante la distanza, è riuscito a starmi vicino lungo tutto il percorso, aiutandomi nei vari passaggi del progetto. Vi ringrazio anche per il supporto nei momenti di sconforto, quasi come psicologi, quando ero tentato di cambiare argomento e abbandonare tutto.
Un ringraziamento speciale va ai miei genitori, Luigi e Lorenza per avermi dato la possibilità di essere qui oggi, sostenendomi in ogni momento e con ogni mezzo a vostra disposizione, comprese le chiamate giornaliere. Non posso non menzionare mio fratello Domenico e mia cognata Giusi, che da sempre mi sostengono nella realizzazione dei miei progetti, offrendo consigli e affetto indispensabili.
Ringrazio di cuore i miei zii che, pur a distanza, hanno sempre un pensiero per me.

Ringrazio Miscioscia, che ha contribuito a mantenere il mio equilibrio mentale attraverso Valorant. Ringrazio i membri del gruppo Instagram, che riempiono le mie giornate con i loro meme. Un grazie particolare ad Aldo per i confronti e i consigli, che hanno contribuito ai risultati di questa tesi. Grazie a Dom, la cui amicizia dura ormai da 14 anni. Ringrazio Teres, che mi fa sentire sempre vicino a Corato, aggiornandomi su tutto. Grazie a Mariano per le conversazioni che condividiamo ogni volta che ci vediamo. Un grazie a Giulia, Fabrrrr e Franco per l'ultima vacanza, che non dimenticherò facilmente. Grazie a Caterina e Fabrizio per la loro sincera amicizia. Ringrazio anche Susanna, che mi permette di visitare almeno una volta all'anno il castello. Un pensiero va anche a tutte le persone che ho incontrato a Pavia, che hanno arricchito la mia vita. In particolare, ringrazio Edoardo, con cui ho condiviso molte perplessità e preparato tanti esami, sin dal primo; Roberta, per la sua costante disponibilità a fare due passi insieme; e Giovanni, con cui ho confronti quotidiani sull'università e che aiuta Miscioscia nella sua missione.
Infine, un grazie di cuore a Giuseppe Adduci, che mi è stato vicino in ogni modo e in ogni momento, specialmente nelle ultime settimane. Senza il suo supporto, tutto sarebbe stato molto più difficile, se non impossibile. Ringrazio, quindi, tutte le persone che mi sono state vicine, anche solo con un saluto, rendendo possibile questo viaggio che ha avuto inizio al liceo e mi ha condotto fino a qui.