Archie A HARKNESS
Matriculation number S1606384

BEng Hons Project Report

Electronic Phantoms for Producing Realistic Medical Images

28 April 2021

Mission Statement

Project Definition

The aim of this project is to use electronic phantoms which mimic the human body alongside the physics simulator program, GATE, to generate realistic CT scans and X-ray images. The data collected from these scans can be useful for training machine learning algorithms to assist with image enhancement, disease detection and diagnosis. Collecting large amounts of data is important when training these machine learning algorithms but the ethical concerns associated with subjecting real people to countless CT scans for this purpose are a real problem due to the fact that the ionising radiation produced during X-rays can cause structural damage to human tissue and organs. By collecting data through computer generated simulations, these ethical concerns are removed, allowing for vast quantities of data to be collected.

Preparatory Tasks

- Research the theory behind the production of X-rays and CT scans
- Learn how these scans can be interpreted in a clinical setting
- Gain an understanding of the importance of the analysis of these scans in diagnostic medicine and the advantage of integrating machine learning algorithms to assist physicians with interpreting data produced from the scans
- Gain necessary programming knowledge to assist in my understanding of the software involved in this project

Interim Targets

- Get GATE software running
- Become familiar with how the program operates
- Produce X-ray simulations using electronic phantoms

- Ideally, I would be able to set up a simulation for producing CT scans by the end of the semester in preparation for collecting CT data in semester 2

The supervisor and student are satisfied that this project is suitable for performance and assessment in accordance with the guidelines of the course documentation.

Signed

Student: ....

University Supervisor: ....

Date: ....

Abstract

In order to train machine learning algorithms, large quantities of data are required. This is a problem due to the fact that images obtained from a CT scan require a patient to be exposed to ionizing radiation, which can be damaging to their health. Since repeated, unnecessary scans of an individual can be harmful and therefore not at all ethical, new methods for collecting medical images in large quantities must be devised. With the emergence of computer physics simulation software, such as GATE, various medical images can be constructed with the only limitation being the computation time for simulation. The simulations in this project are carried out using a very detailed electronic human phantom, called 'Female Adult Mesh' (FASH) which has a more than sufficiently realistic anatomy for the purposes of CT simulation. The software, GATE, was used in this project as it can perform highly sophisticated simulations and does not require any prior C++ programming knowledge; instead using its own scripted macro language which has been designed to make the programming reasonably straightforward [1]. In this thesis, I explain the process involved in producing x-ray images and CT images using the software GATE, MATLAB and ROOT [2]. I also explain methods that can be used to accelerate simulations by splitting the simulation into several macro files which can be run in parallel across several CPUs using Condor. Most importantly, I attach the necessary code that can be used to create reproducible CT and x-ray images, with the idea being that these images can be used in combination with machine learning algorithms in future projects.

This project has proven to be successful as it has produced medical images with life-like artefacts for a fan-beam CT geometry.

Declaration of Originality

I declare that this thesis is my
original work except where stated.

....

Statement of Achievement

For this project, I carried out the background research into the theory behind X-ray and CT scanners. I also learned the necessary skills to use programs such as GATE, ROOT and MATLAB for producing my results. After initial research, I set up several simulations using GATE for technologies including FBCT and X-ray scanners. I used MATLAB and ROOT to analyse the simulation outputs and create x-ray projection images, sinograms and reconstructed CT images.

Contents

List of Symbols

I_0	Initial x-ray intensity.
I_x	Intensity of transmitted x-rays.
λ	Wavelength.
μ	Linear x-ray attenuation coefficient.
c	Speed of light.
h	Planck's constant.

Glossary

ART **A**lgebraic **R**econstruction **T**echnique.

CT **C**omputed **T**omography.

DICOM **D**igital **I**maging and **CO**mmunications in **M**edicine.

FASH **F**emale **A**dult **M**e**S**h.

FBCT **F**an **B**eam **C**omputed **T**omography.

FBP **F**iltered **B**ack-**P**rojection.

GATE **G**eant4 **A**pplication for **T**omographic **E**mission.

GEANT4 for **GE**ometry **AN**d **T**racking.

HU **H**oundsfield **U**nits.

MASH **M**ale **A**dult **M**e**S**h.

MFP **M**ean **F**ree **P**ath.

PE **P**hotoelectric **E**ffect.

PET **P**ositron **E**mission **T**omography.

SPECT **S**ingle-**P**hoton **E**missionComputed **T**omography.

List of Figures

List of Tables

Chapter 1

Introduction

Machine learning is an incredibly powerful tool with a myriad of potential applications within the fields of science and engineering. One particular area of interest for implementation of these algorithms is in the analysis of medical images. Medical images, such as those produced by CT scanners, contain several different types of unwanted artefacts which make it difficult for physicians to identify features of the underlying anatomical data within the scan. It is therefore vitally important that these artefacts are minimised so that key markers for certain medical conditions, such as a malignant tumor indicating cancer, are not overlooked. With early and efficient detection of these markers, the best possible treatment plan can be devised for the patient.

Machine learning algorithms can act as a catalyst for data collection and analysis. Large amounts of data from medical images can be fed into machine learning algorithms, training them to be able to recognise and outline organs and correct any unwanted artefacts.

The issue with acquiring large quantities of medical imaging data is that it is unsafe and unethical to expose patients to excessive amounts of ionising radiation solely for the purpose of data collection. Addressing this concern is the driving force behind the project, in which I hope to produce realistic medical images through the use of highly sophisticated computer simulations. These images will contain realistic artefacts that are typical of those found in real-life scans, so can be used in future projects to train these machine learning algorithms.

The simulation platform that I will use is the program, GATE, which is based on GEANT4 - a physics simulation program. GATE allows simulations to be performed for various types of medical imaging procedures such as PET, SPECT and CT. For this project, I will focus on simulating a fan-beam CT (FBCT) scanner.

The simulations will use an electronic human phantom which contains accurate information about the anatomy and organ densities of a real human. The phantom that will be used for this project is the

'Female Adult Mesh' (FASH) phantom. The advantage of using an electronic phantom instead of a real patient is that an electronic phantom, unlike a real patient, will not move in the duration of the scan. This means that CT images will not contain artefacts caused by random movements such as a heartbeat or breathing.

1.1 Project overview

1.2 Summary

This project will be split into 4 stages:

- The first stage will involve setting up an x-ray simulation in GATE and gathering x-ray images of a basic phantom.
- The next stage will be to modify the simulation geometry in order to run a CT simulation for the basic phantom. This stage will also include the assembly of a sinogram and reconstruction of the CT images from the output files which are produced in GATE.
- Next, the human phantom will be converted into a DICOM file type to be imported into the x-ray simulation. The geometry of this simulation will be adapted for the use of the human phantom and x-ray projections will be gathered.
- Lastly, The full CT simulation will be run for the human phantom to produce CT images containing artefacts which commonly appear in real CT images.

Chapter 2

Background

2.1 X-rays

2.1.1 The theory of X-rays

X-rays are high frequency ionising electromagnetic radiation. They can be produced when the electrons emitted by the cathode filament of an x-ray tube are accelerated, using a high voltage, towards the anode. The electrons then interact with the atoms of the target anode material to produce x-rays.

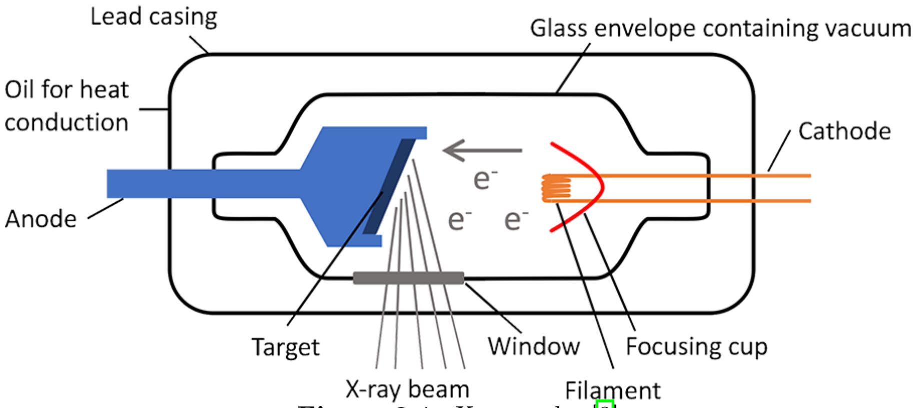


Figure 2.1: X-ray tube [3]

The first type of x-rays that can be produced are called bremsstrahlung, or braking radiation, and are produced when many electrons strike the anode and encounter probabilistic interactions with the material's atoms which cause the electrons to decelerate, producing x-ray photons. Although x-rays are produced, most of the energy from the incident electrons will be converted into heat therefore it is important that an appropriate material is chosen for the target - a common target material used in x-ray tubes is tungsten as it has a good efficiency for x-ray production at high temperatures. Bremsstrahlung x-rays are are polyenergetic, meaning that photons produced will have a wide energy distribution. This means that low energy photons must be filtered at the source since they will not be able to penetrate through a patient's body and reach the detector. Unless low energy photons are filtered out (i.e. by a sheet of metal), these photons would unnecessarily increase the patients absorbed radiation dose without actually contributing to the CT scan in a meaningful way [4].

The second type of x-rays produced are called 'characteristic x-rays' which, as the name suggests, yields a spectrum which is characteristic of a specific anode material. In figure 2.2, this can be observed as the two spikes in the histogram. The smooth, curved, part of this histogram is as a result of bremsstrahlung x-rays.

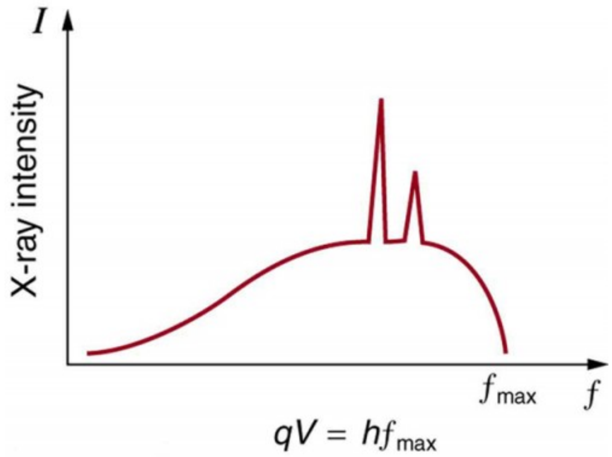


Figure 2.2: Typical x-ray source histogram showing Bremsstrahlung and Characteristic x-rays [5]

2.1.2 X-ray attenuation

X-ray photons have the potential to interact with a material in a few different ways:

- Rayleigh scattering
- Compton scattering
- Absorption

Absorption
One possible fate of incident x-rays is that they will be absorbed by the atoms of the material through which they are passing, causing the emission of a photoelectron. The process known as the photoelectric effect (PE). When photoelectric absorption occurs, an x-ray photon transfers its energy to an atom's inner most (K-shell) electrons, allowing them to break free from the nucleus. This electron is now referred to as a photoelectron. The vacancy in the electron shell will then be filled with a higher energy level

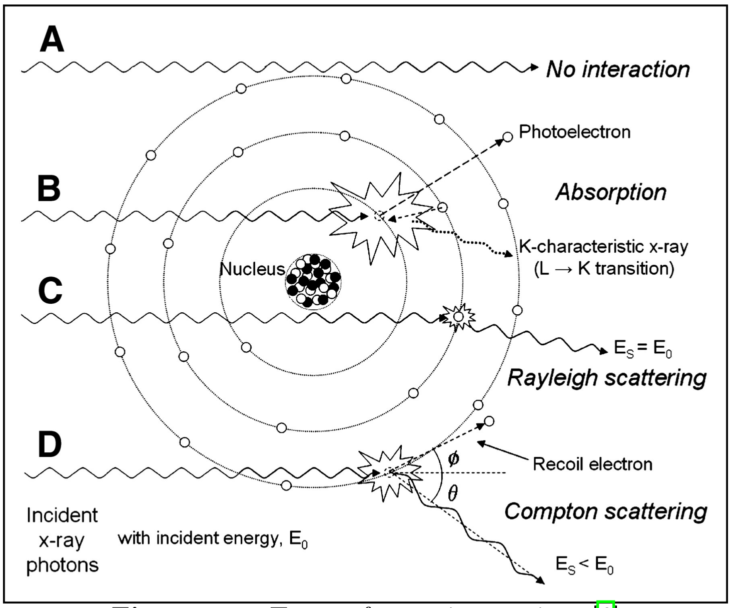


Figure 2.3: Types of x-ray interactions [6]

electron, causing the release of what are called characteristic x-rays.

Compton Scattering

Compton scattering is the main cause of radiation scatter in a material and occurs when the x-ray photon interacts with free electrons or loosely bound electrons in the outer (valence) shell of an atom [7]. These types of interaction cause the photons to scatter and transfer some energy to the electron. The scattered photons will have a a different wavelength after the interaction and thus a different energy given by the following equation:

$$E = \frac{hc}{\lambda} \tag{2.1}$$

Where E is the photon energy, h is planck's constant, c is the speed of light and λ is the wavelength.

Rayleigh Scattering

Rayleigh Scattering typically occurs for lower energy photons in the range of 15 to 30keV. This type of scattering happens when the incident photon does not lose any energy upon interaction with the atoms of a material but has its trajectory slightly altered from what it was before the interaction [8].

Scattering can cause inaccuracies in the final CT image. This is due to the fact that scattered photons can hit the detector in an unexpected position - their initial trajectory has been modified by the materials through which they pass. Since CT imaging typically uses higher energy photons for scanning procedures, Rayleigh scattering does not normally cause a significant amount of distortion in the CT image.

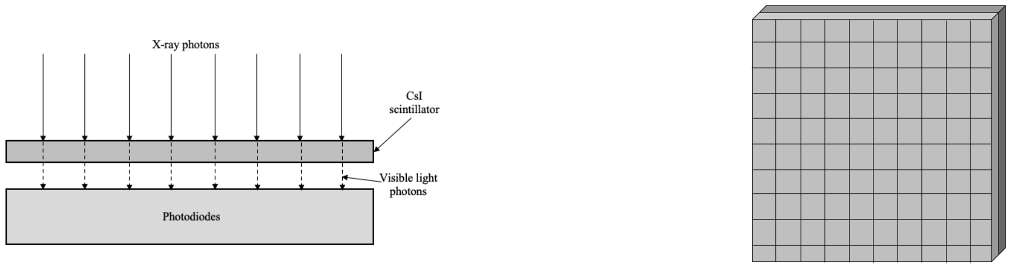
The total attenuation experienced by the x-ray photon can therefore be expressed by the following equation:

$$\mu = \mu_{Compton} + \mu_{Rayleigh} + \mu_{Photoelectric} \tag{2.2}$$

Where μ is the total attenuation coefficient and $\mu_{Compton}$, $\mu_{Rayleigh}$ and $\mu_{Photoelectric}$ are the attenuation coefficients for the different physical interactions that the x-ray photon may experience.

2.1.3 X-ray detection

An x-ray detector typically consists of a 2D array of pixels, positioned behind the object being scanned (with respect to the x-ray source). In order to detect the intensity of x-ray photons at the detector, they must first be converted into visible light photons through the use of a cesium iodide (CsI) scintillator layer on the surface of the detector, as shown in Figure 5.11a. The intensity of the visible light photons can then be measured by the photo-diodes which make up the pixels within the detector and then converted into a digital signal.



(a) X-ray detector

(b) 2D array of pixels which make up the detector

Figure 2.4: Structure of an x-ray detector panel

The Beer-Lambert Law describes the relationship between the detected x-ray beam intensity and the total attenuation coefficient:

$$I_x = I_0 e^{-\mu x} \tag{2.3}$$

Where I_x is the intensity of the transmitted x-rays, I_0 is the initial intensity, μ is the linear attenuation coefficient and x is the thickness of the material.

2.2 Computed Tomography

Computed tomography (CT) was invented in 1772 by Sir Godfrey Hounsfield [9] and since then, this technology has become an incredibly popular and valuable technique for producing 3D medical images. This technology works by rotating an x-ray source and detector around the patient, who will usually be lying down on a bench, and collecting x-ray projection data for every rotation. This x-ray projection data can then be manipulated through a mathematical technique called a Radon Transform to produce the 2D reconstructed CT image which allows an axial slice of the patient to be observed. If enough data is collected, these 2D CT slices can be stacked to form a 3D representation of the scanned object. The data collected from CT scans is commonly used in diagnostics and in monitoring the progression of diseases such as cancer in order to provide the best treatment plan.

Although highly effective, CT scans are usually only offered when deemed completely necessary due to the fact that the ionising radiation that passes through the body during the scanning procedure can cause damage to otherwise healthy tissue. For this reason, efforts are always being made by the scientific community to provide reduced dose CT scans.

2.2.1 Fan-beam CT

The most common type of CT scanner is the fan-beam CT (FBCT) scanner. This type of CT scanner uses an x-ray source which emits a thin fanned beam of x-rays. A FBCT scanner consists of several components: an x-ray source, detector array, collimator and filter.

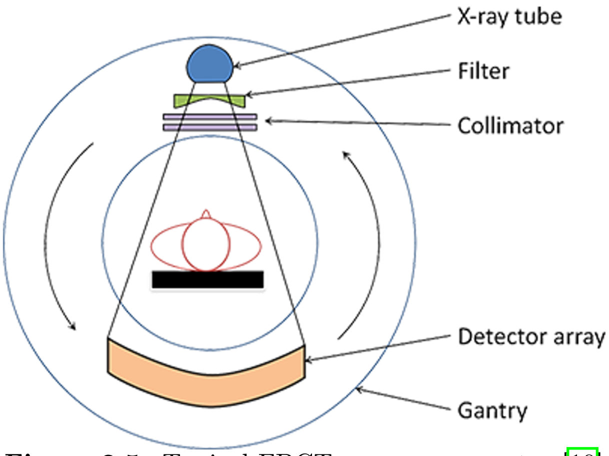


Figure 2.5: Typical FBCT scanner apparatus [10]

The x-ray source for a FBCT scanner typically has a peak electron energy of 120keV as it provides an optimum contrast in the resulting CT image for distinguishing between the different structures within the body. A tungsten anode is also very commonly used for this type of CT scanner.

The detector for a FBCT scanner is a linear or curved array of pixels. A scintillator on the surface of the detector converts the x-ray photons into visible light photons. Again, the photo-diodes in the detector can then convert this visible light signal into an electrical signal that can then be digitised.

A piece of equipment called a collimator is placed between the x-ray source and the patient. This serves the purpose of flattening the x-ray beam from a cone-beam into a fan-beam distribution. By narrowing the x-ray beam, unnecessary exposure of the patient is reduced for x-ray photons that would never have reached the detector. A collimator is usually constructed out of a very dense material, such as lead, in order to effectively attenuate the unwanted incident x-ray photons.

There is a sheet of metal, often aluminium, which acts as filter between the x-ray source and the patient which shapes the x-ray source spectrum so that lower energy photons are removed before reaching the patient. A 'bow-tie filter' is often used to even the attenuation experienced by the x-rays as they pass through the patient. This will reduce artefacts in the CT image that are produced as a result of 'beam hardening' - a concept which will be explained shortly.

2.2.2 Sinogram and Radon Transform

The Radon Transform is a mathematical tool that relates a 2D function f(x, y) to the line integrals of that function. In other words, it is a function that allows us to determine the structure inside the body from a set of linear projections through it [11]. This technique is shown in Figure 2.6. The reason why the Radon Transform is useful in the construction of CT images is that a series of, essentially, line integrals are produced by measuring the level of attenuation that x-rays experience as they pass through an object, at intervals spanning at least 180 degrees of rotation. By performing an inverse Radon Transform on a sinogram, the reconstructed CT image is produced.

The expression for finding the Radon Transform of a 2D function is given by equation 2.4.

$$Rf(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\delta(xcos\varphi + ysin\varphi - r)dxdy \tag{2.4}$$

The sinogram can then be transformed into a reconstructed CT image through the process of back-projection. As mentioned in Chapter 2, various other techniques for image reconstruction exist, back-projection being the most straightforward. A filter can be applied to the back projected image to obtain what is called a filtered back-projection (FBP).

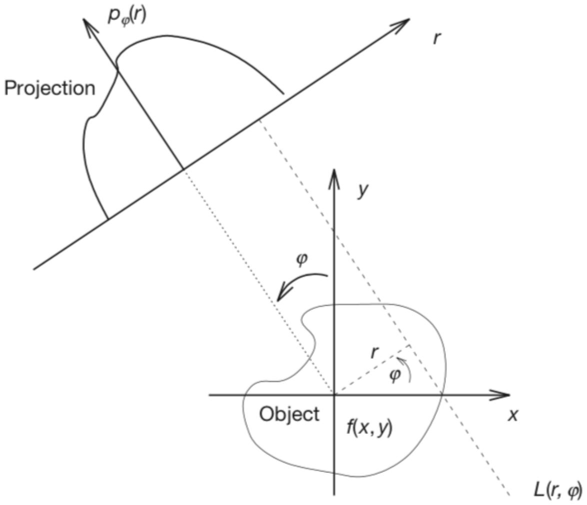


Figure 2.6: Radon Transform theory [12]

The resulting CT image is a grey-scale image where different shades correspond to different densities of scanned material [13]. A way of relating the level of attenuation experienced by the x-ray photons to the grey-scale value can be described by Houndsfiled Units (HU).

$$HU = 1000 \times \frac{\mu - \mu_{water}}{\mu_{water} - \mu_{air}} \tag{2.5}$$

Where μ_{water} is the attention coefficient of distilled water at standard temperature and pressure. Some examples of HU values for different materials are included in table 2.1.

Material	HU
Bone	+1000
Soft tissue	+40
Water	0
Fat	-100
Air	-1000

Table 2.1: HU for different materials [14]

2.2.3 Reconstruction

CT reconstruction is the process of converting the CT image data from a sinogram into a 2D axial slice of the patient. 180 degrees of projection data are normally required to form a reconstructed CT image - going beyond 180 degrees would not be useful because you would essentially be producing duplicate images as the projection of the phantom will look like a mirror image of itself after 180 degrees. There are a few different methods of CT reconstruction including: back-projection, filtered back-projection (FBP) and iterative reconstruction.

Back Projection: The simplest form of CT image reconstruction is carried out through a 'back projection'. This method involves taking x-ray projections for each angle of acquisition and "smearing" the projection's attenuation data onto the image at the angle from which it was taken [15]. This will produce a reconstructed image although it will be blurry and the finer, more useful, features of the image are obscured.

Filtered back projection: A filtered back-projection is similar to the back-projection with the exception being that a filter is applied to the sinogram image before the reconstruction process takes place. This corrects the blurriness of the simple back-projected image, allowing a higher level of detail to be observed.

Iterative reconstruction: Iterative reconstruction is a relatively new CT image reconstruction method as it requires more advanced computational processes than the previous methods. This method has the advantage that it can overcome some of the noise that will be visible in methods such as FBP. The iterative reconstruction algorithm usually begins by making an assumption about the CT image and as the reconstruction process progresses, this assumption is refined [16]. The reconstructed image and the assumed image are compared until both images appear to be identical.

2.2.4 Artefacts

Artefacts in CT images can degrade the quality of the image and potentially hide important features of the scan. There are several common types of imaging artefacts that are observed in a typical CT scan. The most common artefact types can be defined in one of the following ways:

- Beam hardening artefact
- Photon starvation
- Motion artefact

Beam hardening artefacts are often the most common type of artefact observed in CT images [17]. Poly-energetic x-ray sources produce x-ray photons with a wide range of energies. When these x-rays pass through a material, the mean energy of the photons increases as lower energy photons are attenuated. The beam is said to have become 'harder' as a result of this effect. A type of beam hardening artefact called 'cupping artefacts' occur as a result of x-rays passing through a cylindrically shaped object of uniform density. The photons passing through the middle of the object are more 'hardened' compared to those passing through the edges of the object. Figure 2.7 shows some typical features of a CT image

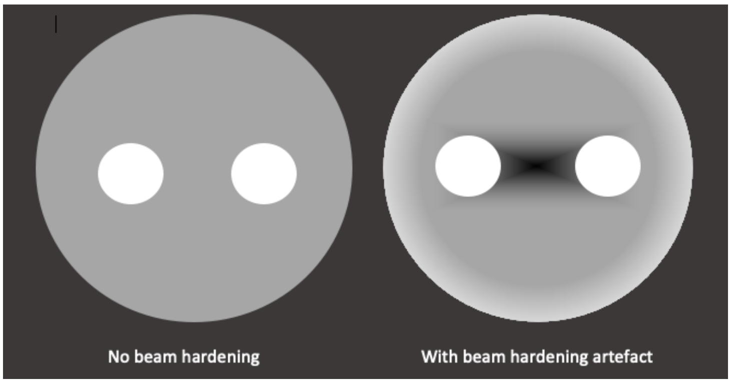


Figure 2.7: Example of artefacts caused by beam hardening

which contains distinct beam hardening artefacts. Cupping artefact can be observed as a bright gradient towards the circumference of the scanned object.

Photon starvation artefacts can be observed behind areas of high x-ray attenuation in reconstructed CT images. This type of artefact arises due to the fact that an insufficient number of photons reach the detector because of the high attenuation from materials such as metal implants. This artefact can be seen as streaking in the image.

Motion artefact arises due to the random movements of the patient in the duration of the CT scanning procedure. These movements can be due to uncontrollable factors such as breathing or a heartbeat or simply due to the patients general restlessness during the scan. As mentioned before, this type of artefact can be avoided completely through the use of an electronic phantom such as the FASH or MASH phantoms.

2.3 FASH/MASH Phantoms

For this project, an accurately modelled human phantom will be used to produce realistic medical images. These phantoms contain accurate information with regard to organ and bone densities as well as the geometry of all of the body's internal and external structures.

The Female Adult MeSH (FASH) and Male Adult MeSH (MASH) are voxellised[1] phantoms that are modelled to a very high degree of accuracy to represent the anatomy of the human body [18]. These phantoms were created using software such as Blender, MakeHuman, Binvox and ImageJ through which organ masses had been carefully selected to be of the values recommended by the International Commission on Radiological Protection for the male and female reference adult in report No. 89 [19].

[1] Voxels are the 3D equivalent of pixels

The FASH phantom will be used in this project. In particular, the FASH phantom modelled to be in a supine position, to accurately simulate the CT scanning procedure, in which the patient is lying on their back. The main difference between the standing and supine phantom is that the supine phantom's organs are shifted slightly due to the effects of gravity.

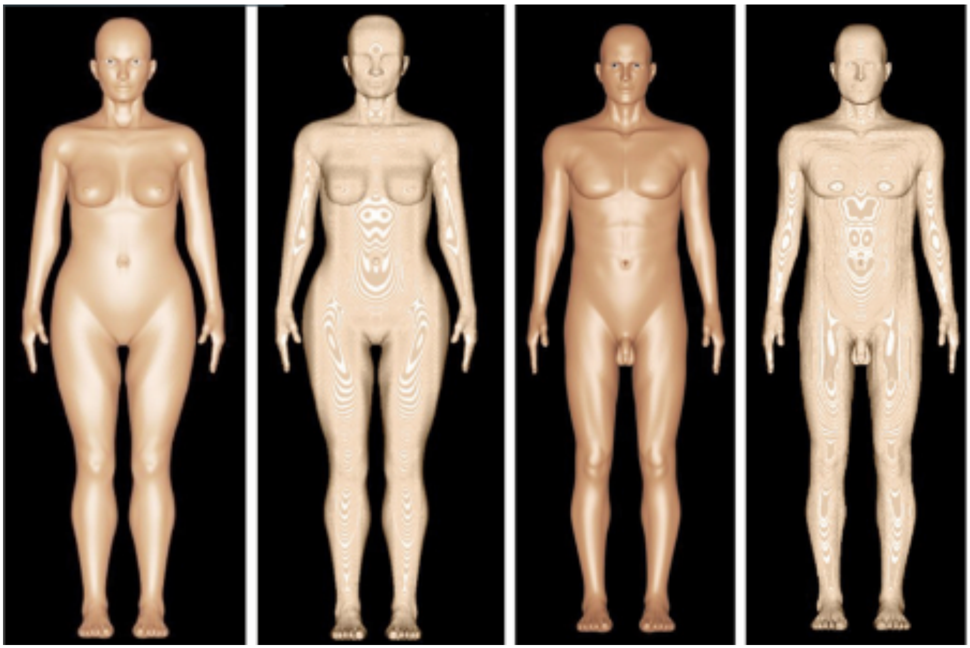


Figure 2.8: FASH and MASH 3D model and voxellised phantoms

2.4 GATE

For this project the software, GATE, will be used. GATE is a Monte Carlo simulation program based on the (GEANT4) simulation toolkit which contains well validated physics models [20]. GATE allows medical imaging procedures to be accurately simulated to produce realistic imaging data for a variety of scanner types and geometries. The geometry of these simulations can be highly customised in order to replicate the real-life procedure in detail.

Monte Carlo Simulations

Monte Carlo simulations use randomness to solve deterministic problems [21]. Any sort of chance event, such as the way in which x-ray photons scatter and attenuate upon interacting with a material, can be accurately modelled using this types of simulation.

User Interface

GATE requires no C++ programming as it operated through the use of a dedicated scripting mechanism, referred to as a macro language, which makes the setting up of the simulations more straightforward.

The user must enter script to the command line to set up the simulation or alternatively, macro files containing script can be called by the program in order to use predefined geometry and physics processes.

Output

GATE has two different output file types for CT simulations: ROOT and 'ImageCT'. The first type of output file is a ROOT file. ROOT is a program that is used for analysing and plotting data about the positions of particles in the detector during the simulation.

The ImageCT output file is a binary matrix of float numbers that contains information about the number of 'singles' for each time slice. Singles are hits that are registered and digitized within the detector. The ImageCT file can be read into MATLAB where matrix manipulations can be carried out for assembling sinograms and reconstructed CT images.

Chapter 3

X-ray Simulation

This chapter will describe the process of setting up a basic x-ray simulation in GATE. It will then go on to explain how the simulation output files can be read and interpreted by MATLAB and ROOT.

3.1 Simulation setup

3.1.1 Macros

As mentioned previously, a macro language is used to interact with GATE. Macro files containing script can be created and called into the program to make the simulation setup more efficient. The simulations require several macro files to describe the different aspects of the simulation, such as:

- the physics processes involved
- the scanner geometry
- the number of photons produced by the x-ray source per acquisition
- the output file type
- the phantom geometry
- the x-ray source

Organising the simulation code into separate macro files in a modular way makes it easier to modify the simulation in-between runs.

3.1.2 Simulation Geometry

The first aspect of the simulation geometry that must be established is the world volume. The 'world' has a box geometry and contains all of the simulation's secondary volumes. The centre of the world volume is referred to as the isocentre and is where the phantom will be located in the simulation. The x-ray detector, phantom and source will all be placed within the world volume. The following script provides an example of how the world volume is defined in GATE:

```
# World
/gate/world/geometry/setXLength 500. mm
/gate/world/geometry/setYLength 500. mm
/gate/world/geometry/setZLength 500. mm
/gate/world/setMaterial Air
/gate/world/vis/forceWireframe
/gate/world/vis/setColor black
```

The x-ray detector consists of a two dimensional array of pixels positioned 80mm away from the isocentre. GATE has a predefined system called CTscanner that contains information about how particles interact with the detector. The following script shows how the x-ray detector can be defined using this system:

```
# CTscanner system
/gate/world/daughters/name CTscanner
/gate/world/daughters/insert box
/gate/CTscanner/geometry/setXLength  60. mm
/gate/CTscanner/geometry/setYLength  60. mm
/gate/CTscanner/geometry/setZLength   3. mm
/gate/CTscanner/placement/setTranslation 0. 0. 80. mm
/gate/CTscanner/setMaterial Air
/gate/CTscanner/vis/forceWireframe
/gate/CTscanner/vis/setColor white
```

Simulation parameter	Value
World size	$500 \times 500 \times 500mm$
Phantom	Water cylinder
Pixels in Detector	200×150
Source to detector distance	160mm
Detector to isocentre distance	80mm
Source kVp	40keV

Table 3.1: Basic x-ray simulation parameters

3.1.3 Source

An x-ray source is placed 160mm from the detector. The source produces x-rays in a cone beam shape which pass through the phantom and hit the detector. The spectrum of the source must be defined by the user. The x-ray source spectrum was chosen here to have a range between 10 and 40keV.

3.1.4 Simulation Physics

Different physics processes must be defined within GATE to be simulated. The following lines of script show, as an example, the initiation of some physics processes:

```
/gate/physics/addProcess PhotoElectric
/gate/physics/processes/PhotoElectric/setModel StandardModel

/gate/physics/addProcess Compton
/gate/physics/processes/Compton/setModel StandardModel

/gate/physics/addProcess RayleighScattering
/gate/physics/processes/RayleighScattering/setModel PenelopeModel
```

The physics processes which were defined for running CT and x-ray simulations were: Compton scattering, Rayleigh scattering, photoelectric effect, electron ionisation, bremsstrahlung and multiple scattering.

3.1.5 Digitizer

The digitizer acts as an interface between the signal that is produced by incident x-ray photons on the sensitive detector and the information that represents this interaction on a computer system. A threshold can be set for the minimum energy that the digitizer will detect in order to make the simulation more realistic. For the simulations that I run here, a 10keV threshold is set.

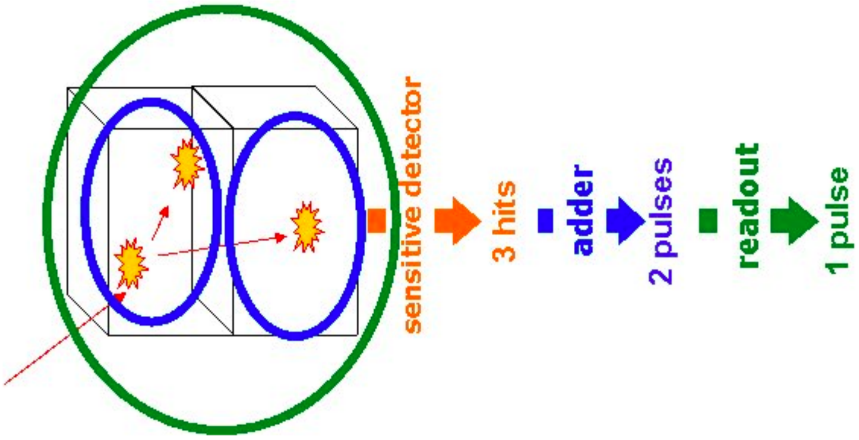


Figure 3.1: Diagram of adder and readout modules from GATE users guide [20]

```
/gate/digitizer/Singles/insert adder
/gate/digitizer/Singles/insert readout
/gate/digitizer/Singles/readout/setDepth 2
/gate/digitizer/Singles/insert thresholder
/gate/digitizer/Singles/thresholder/setThreshold 10 keV
```

Listing 3.1: GATE script to configure digitizer and set 10keV threshold

3.1.6 Material Database

The materials to be used in the simulation must be described in a separate file, containing information such as the material name, its density and its elemental composition.This information was added to a file which I named 'MyMaterialDatabase.db'.

```
SpineBone: d=1.42 g/cm3 ; n=11
        +el: name=Hydrogen   ; f=0.063
        +el: name=Carbon     ; f=0.261
        +el: name=Nitrogen   ; f=0.039
        +el: name=Oxygen     ; f=0.436
        +el: name=Sodium     ; f=0.001
        +el: name=Magnesium  ; f=0.001
        +el: name=Phosphor   ; f=0.061
        +el: name=Sulfur     ; f=0.003
        +el: name=Chlorine   ; f=0.001
        +el: name=Potassium  ; f=0.001
        +el: name=Calcium    ; f=0.133
```

There is already an existing materials database file in the GATE install directory which describes the different chemical elements and a few different materials. This database can be expanded by the user, allowing different materials to be described.

3.1.7 Phantom

For the basic x-ray simulation, the phantom was chosen to be a cylinder of water with two spheres inside it. One sphere was made with its material set to be aluminium and the other was chosen to be made out of bone. Water was chosen as the cylinder material as the human body is made mostly up of water. Bone and aluminium spheres were used as they were likely to produce visible beam hardening artefacts in the CT simulation.

The phantom was set to rotate by 90 degrees between the first acquisition and the second to output two x-ray projections at different angles for one complete simulation.

3.2 Output

3.2.1 ROOT

The program, ROOT, can be used to plot graphs of the positions of particles and particle interactions throughout the duration of the simulation.

It can be observed from Figure 3.2b that the x-ray photons are attenuated along the paths which interact with the cylindrical phantom. The data from the ROOT file can also be used to plot a projection image as shown in Figure 3.3a.

3.2.2 X-ray projection

The 'ImageCT' output allows x-ray projections from GATE simulations to be plotted in MATLAB. Simulations were run with a varying number of x-ray photons, producing projections of varying image quality as shown in 3.4. For each run, there are two projection acquisitions in this simulation, separated by 90 degrees of rotation.

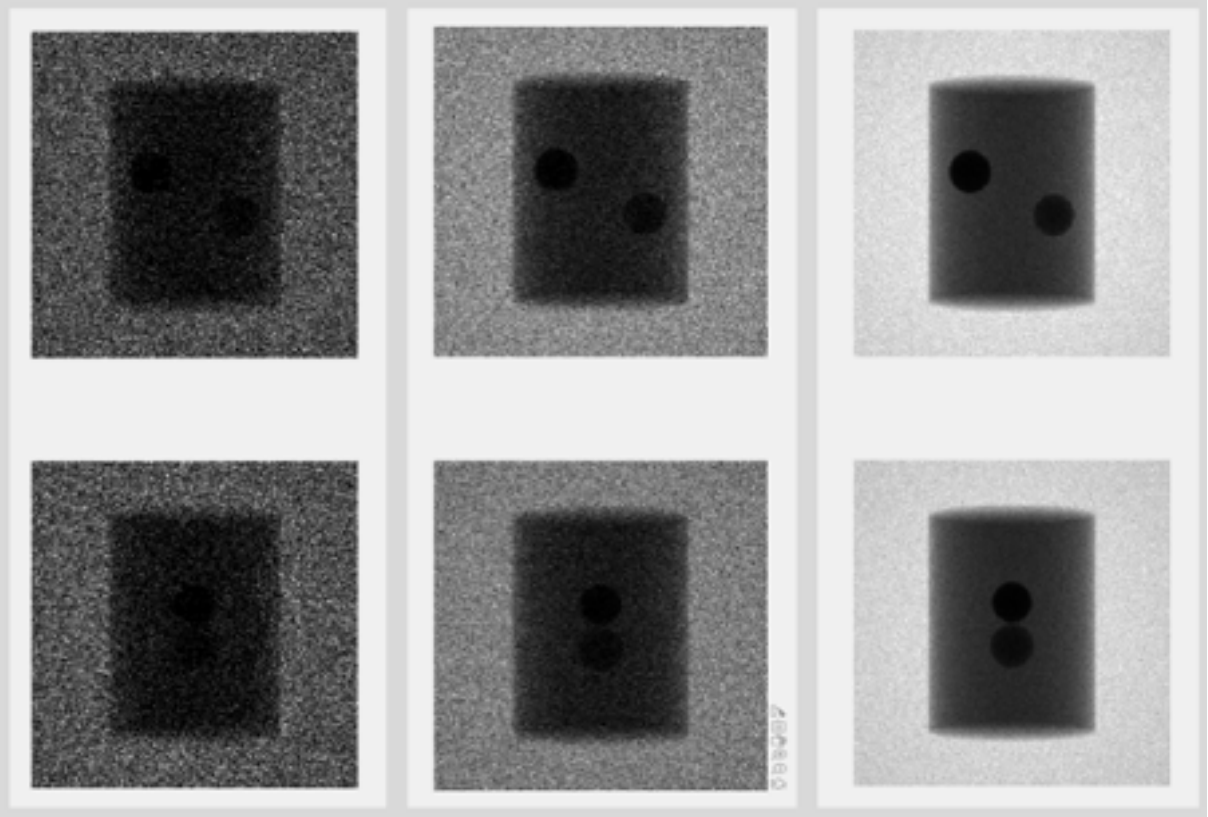
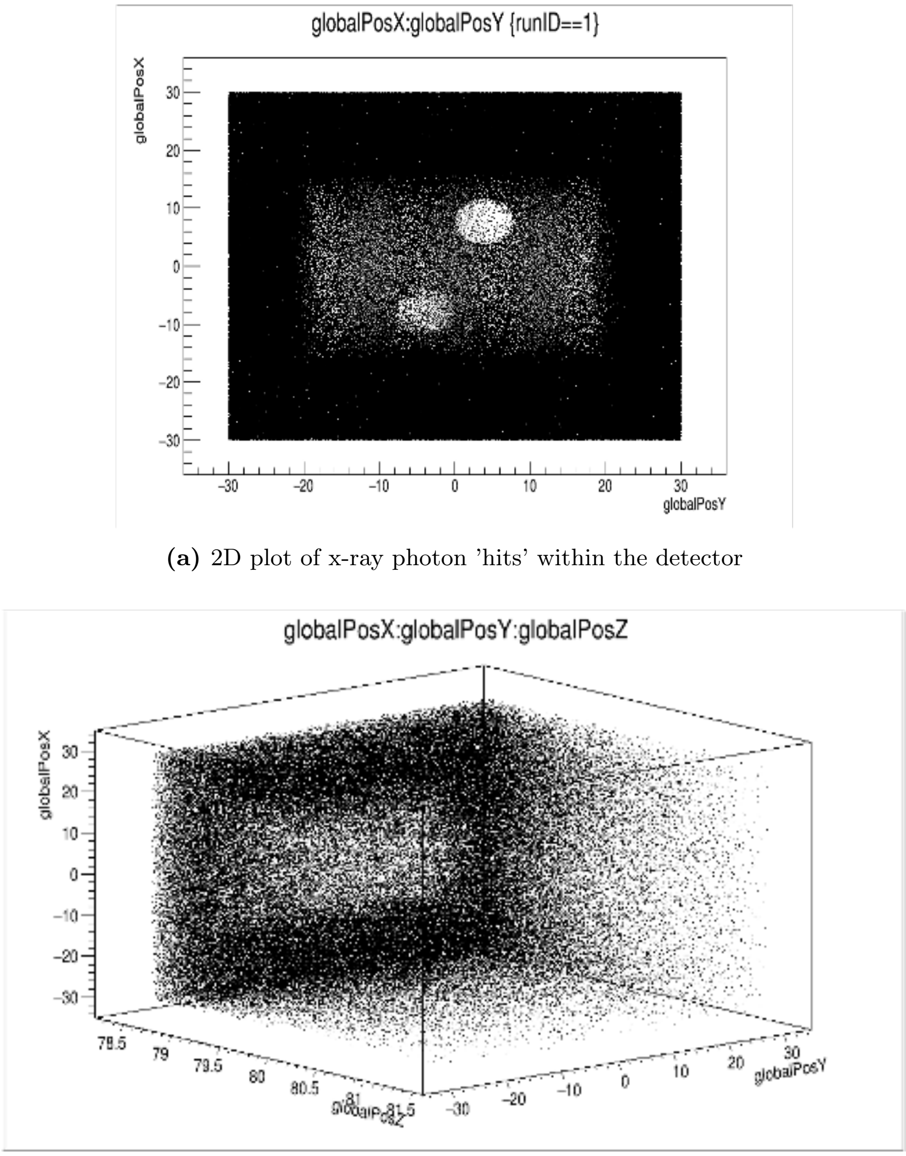


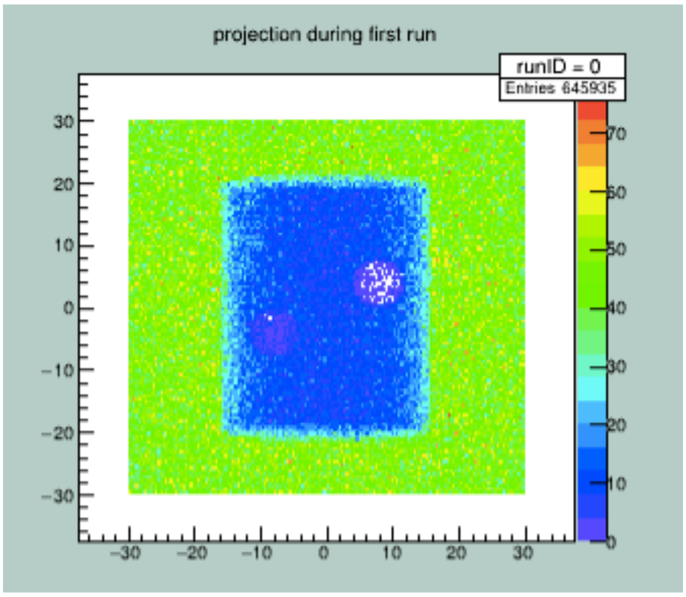
Figure 3.4: X-ray projections for simulations with a) 500,000 b) 5,000,000 photons c) 50,000,000 photons



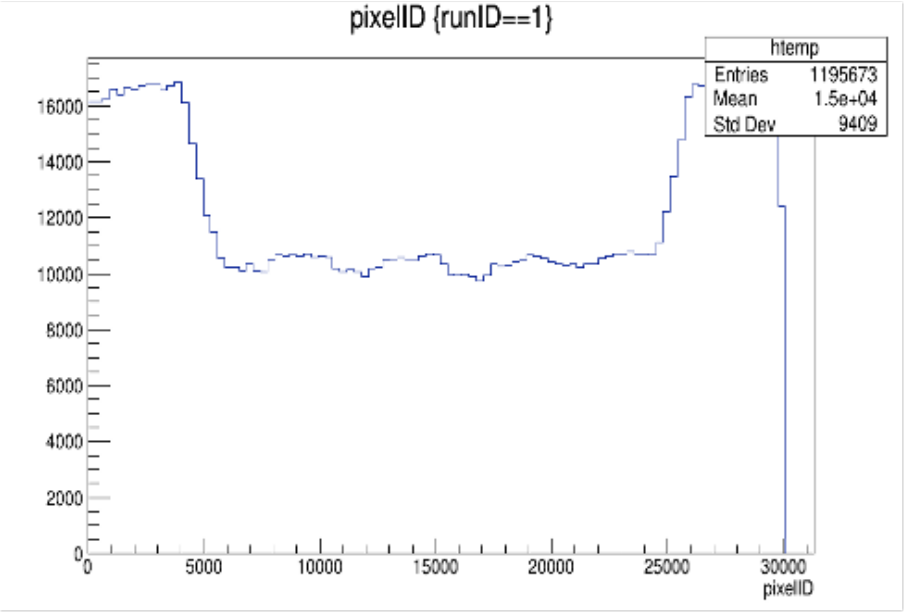
(a) 2D plot of x-ray photon 'hits' within the detector

(b) 3D representation of position of 'hits' within the detector

Figure 3.2: ROOT plots of 'hit' positions inside x-ray detector geometry



(a) X-ray projection image from GATE's ROOT output



(b) Plot of x-ray hits for pixel ID within detector array

Figure 3.3: Graphs plotted from GATE's ROOT output

Chapter 4

CT Simulation

This chapter outlines the method that is used to set up and run a CT simulation in GATE. It then goes on to describe how the simulation output files can be used to create a sinogram and then from that, reconstructed CT images.

4.1 Simulation setup

This section explains how a CT simulation is set up in GATE, using a similar basic phantom geometry to the x-ray simulation example.

4.1.1 Simulation Geometry

For a basic FBCT simulation, the phantom was modified slightly from the basic cylindrical phantom that had been used for obtaining x-ray projections. The cylinder is still composed of water and the spheres are still bone and aluminium, with the only difference being that the spheres are now positioned in the central slice of the phantom, along the x-axis. This is so that the CT image for the central slice of the phantom will be more interesting to look at, with more obvious artefacts visible.

The phantom was set to rotate at a speed of 1 degree/second for a simulation time of 180 seconds. In a real CT scanner, the source and detector rotate around the patient, but here it is more intuitive to simply program the phantom to rotate instead and the effect will be the same.

The detector for a fan beam CT scanner contains a single row of pixels. There are 200 pixels in the detector for this simulation.

A lead collimator was added to flatten the x-ray beams from a cone beam to a fan beam distribution. In practice, this reduces scattering and unnecessary exposure of the patient to radiation. This reduces the simulation time too, as fewer photons have to be tracked through the the world geometry.

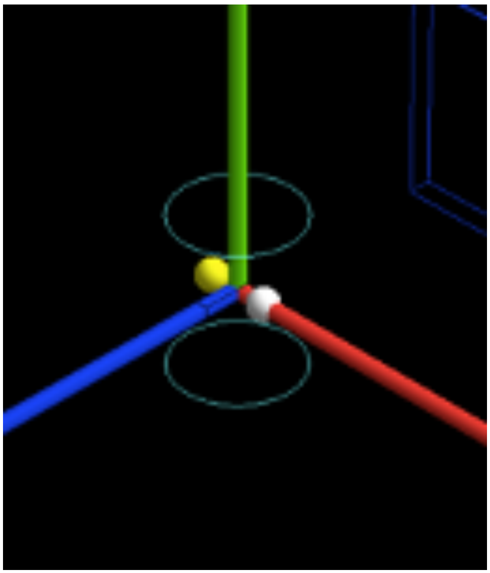


Figure 4.1: Basic phantom for CT simulation

Simulation parameter	Value
World size	$500 \times 500 \times 500mm$
Phantom	Water cylinder
Pixels in Detector	200
Detector to isocentre distance	100mm
Source to detector distance	180mm
Source kVp	40keV
Number of x-ray photons	50 million

Table 4.1: Basic CT simulation parameters

4.2 Output

This section will explain how the ImageCT output file can be used to assemble a sinogram using MATLAB. It will then go on to explain how the sinogram can be transformed into a reconstructed axial CT slice.

4.2.1 Sinogram

Throughout the duration of the simulation, 180 linear projections, each 512 pixels in length are acquired; one for each angle of rotation. Each projection file is saved separately and can be combined using matrix manipulations in MATLAB to produce the sinogram shown in Figure 4.3.Each column of pixels in the sinogram represents a different angle of rotation and, in this case, there are 180 columns in the sinogram which each represent 1 degree of rotation. The contrast of the sinogram was adjusted until the background was black, in order to reduce noise that may appear in the reconstruction.

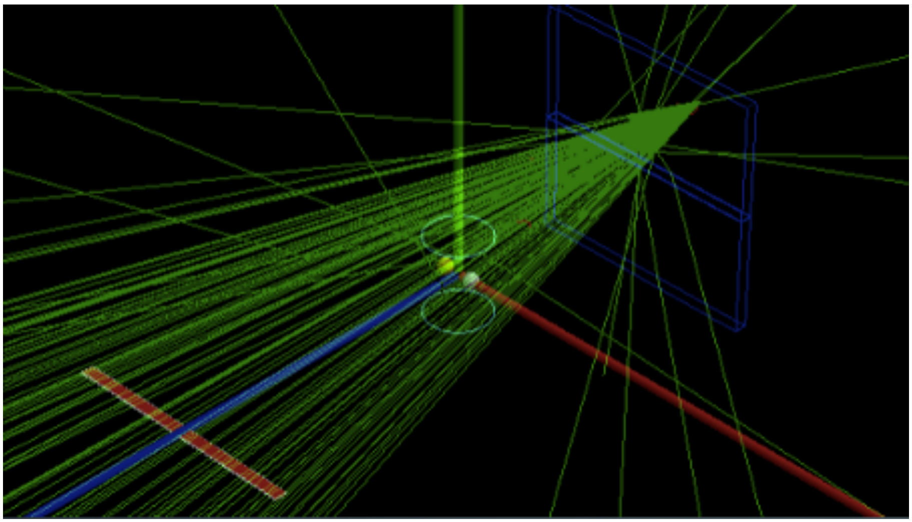


Figure 4.2: Basic FBCT simulation geometry

4.2.2 CT reconstruction

Within MATLAB we can use a useful function, iradon(), which will convert a sinogram into a simple CT back-projection by performing an inverse Radon transform - this is the most basic form of CT reconstruction.The function 'iradon()' also includes a parameter that allows a filter to be applied to the sinogram before the back-projection takes place, thus producing a filtered back-projection (FBP).

A third function was defined in MATLAB which will perform a form of iterative reconstruction called Algebraic Reconstruction Technique (ART)[22].

4.2.3 Discussion

Several clear image artefacts can be observed from these reconstructed CT images. The first and perhaps the most obvious artefact you may notice is the black cross in the centre of the image. This artefact is produced as a result of beam hardening of the x-ray photons. It can also be observed that the phantom has a bright hue around its circumference, which is most likely also a side effect of beam hardening. The beam hardening artefacts at the centre of the CT image are particularly apparent due to the fact that the spheres of aluminium and bone have a high density when compared to the water cylinder. As x-rays pass through areas of higher density, lower energy photons are attenuated, allowing only high energy photons to pass through the material.

Another type of artefact caused by beam hardening that is visible here is the cupping artefact which looks like a bright glow around the circumference of the water cylinder. As mentioned previously, this artefact arises due to the fact that x-ray photons passing through the middle of the object become 'more hardened' compared to those passing through the edges of the object.

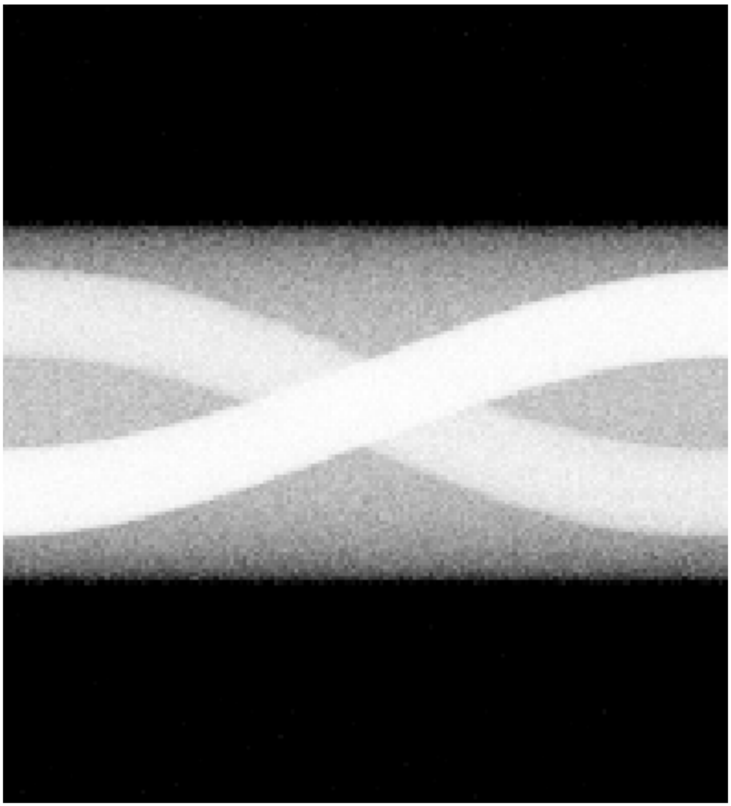


Figure 4.3: Sinogram produced from CT simulation using a basic phantom

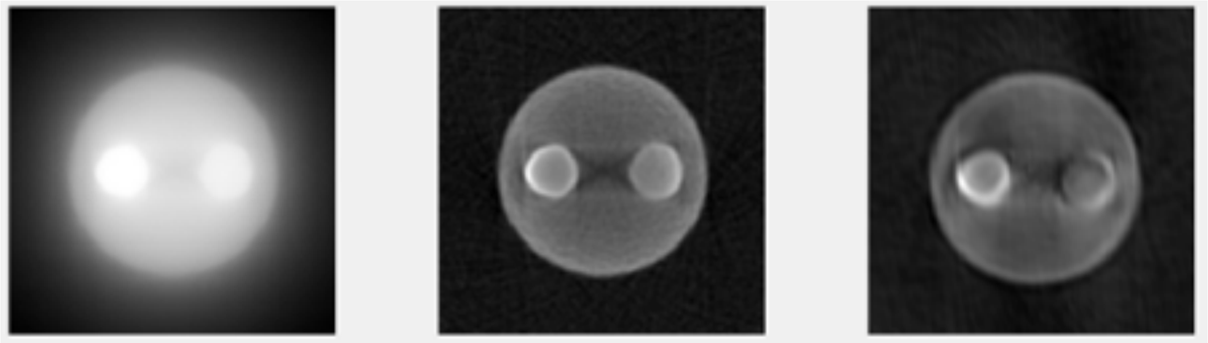


Figure 4.4: a) Back projection b)Filtered back projection c)Iterative reconstraction (ART)

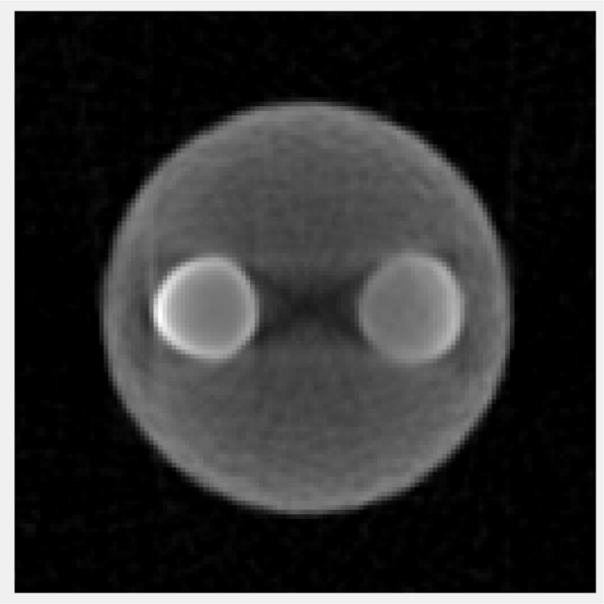


Figure 4.5: Filtered back-projection image of axial slice of the water cylinder phantom with two embedded spheres made of aluminium and bone, respectively.

Chapter 5

FASH/MASH Simulation

The basic CT simulation used previously can be modified to allow for the use of the human phantom in place of the basic cylindrical phantom. The human phantom that will be used here is the supine FASH phantom.

5.1 Simulation setup

5.1.1 Simulation Geometry

As before, a collimator was added to flatten the cone beam shape of the emitted x-ray photons into a fan beam shape. For this simulation, to prevent any unnecessary tracking of particles that would ultimately not take part in the production of the CT image slice, a second collimator was added before the bow-tie filter. There were 512 pixels in the detector, as this is a commonly accepted resolution for ct images [23].

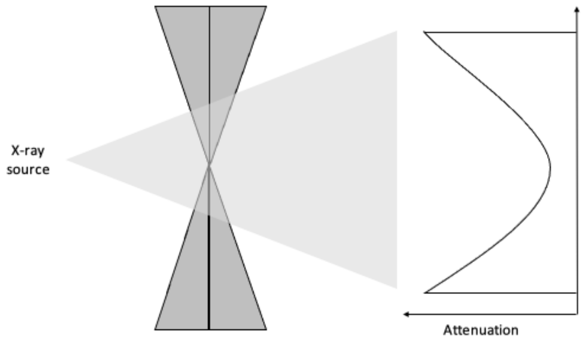


Figure 5.1: This figure shows the level of x-ray attenuation experienced by x-ray photons passing through the bow-tie filter

The addition of an aluminium bow-tie filter was necessary to reduce the dominating beam hardening artefacts in the reconstructed image. X-rays that pass through the centre of a cylindrically shaped phantom experience more beam hardening due to the fact that they must travel through a larger volume of the phantom material.This bow-tie filter hardens the x-ray photons that may experience less attenuation around the circumference of the abdominal phantom. This should reduce any 'haloing' effect that might be observed around the edges of the CT image. The bow-tie filter has been constructed out of 4 aluminium wedges in GATE as shown in Figure 5.1.

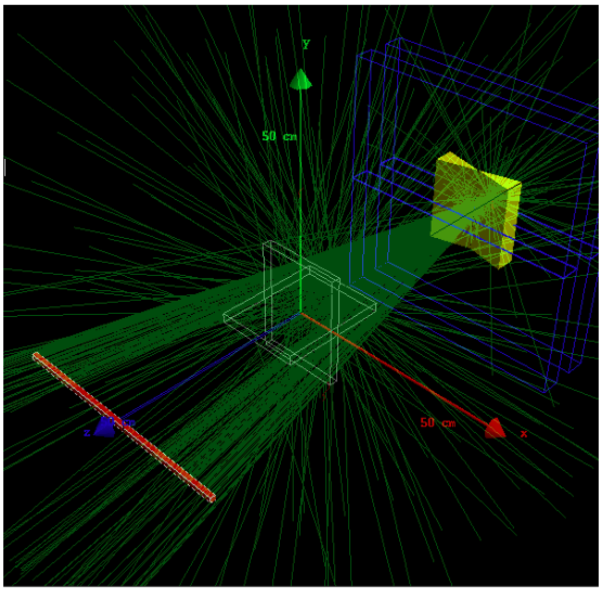


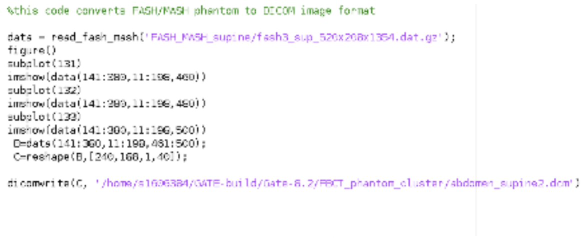
Figure 5.2: Simulation geometry for human phantom

Simulation parameter	Value
World size	$1000 \times 1000 \times 1500mm$
Phantom	FASH
Pixels in Detector	512
Detector to isocentre distance	450mm
Source to detector distance	180mm
Source kVp	120keV
Number of x-ray photons	2 billion

Table 5.1: FASH CT simulation parameters

5.1.2 Defining Phantom Geometry

Phantoms can be defined in GATE that have been imported into the program in DICOM format, a common file type used for medical imaging data. The phantom data was converted to a DICOM file using MATLAB, as shown in Figure 5.3. Since only a small section of the phantom is necessary for running CT simulations for one slice, a section of the phantom was chosen that contained several different and identifiable organs: liver, stomach, gallbladder, spleen and pancreas. Again, the phantom was set to rotate at 1 degree/second for a total simulation time of 180 seconds.



```
%this code converts FASH/MASH phantom to DICOM image format

data = read_fash_mash('FASH_MASH_supine/fash3_sup_520x208x1354.dat.gz');
figure()
subplot(131)
imshow(data(141:380,11:198,460))
subplot(132)
imshow(data(141:380,11:198,480))
subplot(133)
imshow(data(141:380,11:198,500))
 B=data(141:380,11:198,461:500);
 C=reshape(B,[240,188,1,40]);

dicomwrite(C, '/home/s1606384/GATE-build/Gate-8.2/FBCT_phantom_cluster/abdomen_supine2.dcm')
```

Figure 5.3: MATLAB code used to convert FASH/MASH phantoms into DICOM image format

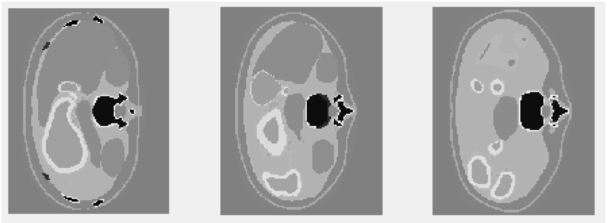
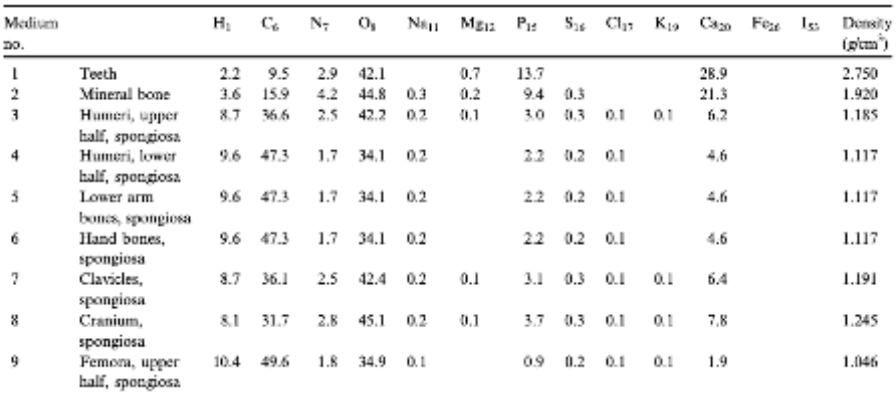


Figure 5.4: Selected slice of FASH phantom used for abdominal CT simulation

Within GATE, the material database had to be updated in order for all of the organ, bone and tissue densities to be defined within the phantom. To do this, the elemental composition and densities of organs were taken from the Annals of the ICRP issue on 'Adult Reference Computational Phantoms' [24].



Medium no.		H_1	C_6	N_7	O_8	Na_{11}	Mg_{12}	P_{15}	S_{16}	Cl_{17}	K_{19}	Ca_{20}	Fe_{26}	I_{53}	Density (g/cm^3)
1	Teeth	2.2	9.5	2.9	42.1		0.7	13.7				28.9			2.750
2	Mineral bone	3.6	15.9	4.2	44.8	0.3	0.2	9.4	0.3			21.3			1.920
3	Humeri, upper half, spongiosa	8.7	36.6	2.5	42.2	0.2	0.1	3.0	0.3	0.1	0.1	6.2			1.185
4	Humeri, lower half, spongiosa	9.6	47.3	1.7	34.1	0.2		2.2	0.2	0.1		4.6			1.117
5	Lower arm bones, spongiosa	9.6	47.3	1.7	34.1	0.2		2.2	0.2	0.1		4.6			1.117
6	Hand bones, spongiosa	9.6	47.3	1.7	34.1	0.2		2.2	0.2	0.1		4.6			1.117
7	Clavicles, spongiosa	8.7	36.1	2.5	42.4	0.2	0.1	3.1	0.3	0.1	0.1	6.4			1.191
8	Cranium, spongiosa	8.1	31.7	2.8	45.1	0.2	0.1	3.7	0.3	0.1	0.1	7.8			1.245
9	Femora, upper half, spongiosa	10.4	49.6	1.8	34.9	0.1		0.9	0.2	0.1	0.1	1.9			1.046

Figure 5.5: List of media, their elemental compositions (percentage by mass), and their densities for the adult female reference computational phantom

5.1.3 Choice of Source kVp

The x-ray source energy spectrum was chosen to have a peak voltage of 120kVp and defined using a program called Spekcalc, which models x-ray sources with a tungsten anode [25], as shown in Figure 5.6. 120kVp is a commonly used source voltage for CT procedures [26], as it produces the best contrast, without causing excessive damage to healthy tissue. Spekcalc allows filtration to be added to the x-ray source histogram. This is beneficial because by removing the need to use filtration methods within the geometry of simulation, the simulations computational complexity would decrease since photons do not need to be tracked through additional filters and thus take less time to run.

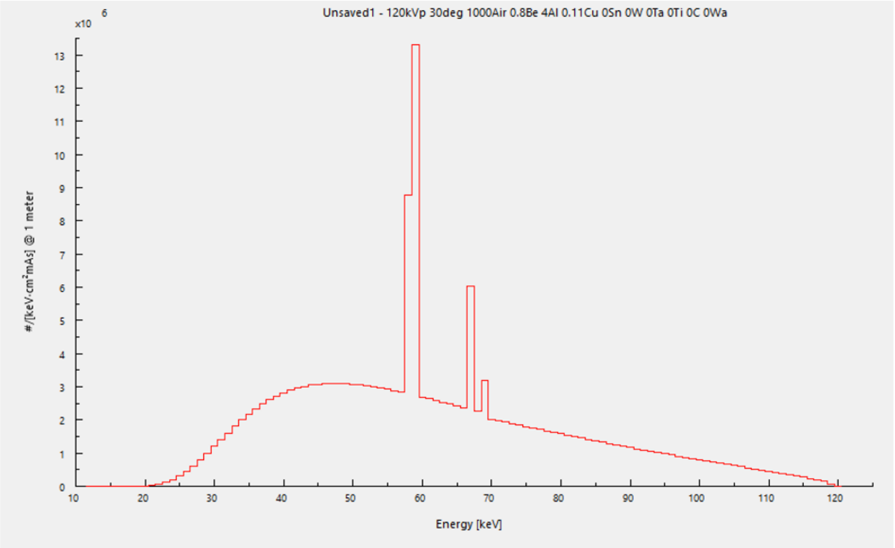


Figure 5.6: 120kVp x-ray source spectrum for tungsten anode and 4mm aluminium filtration

An x-ray image of the head section of the phantom was produced (Figure 5.7) in order to observe the effectiveness of the chosen x-ray source intensity before committing to starting the lengthy CT simulation. It was decided that the images contained a suitable level of detail to go ahead with the full CT simulation.

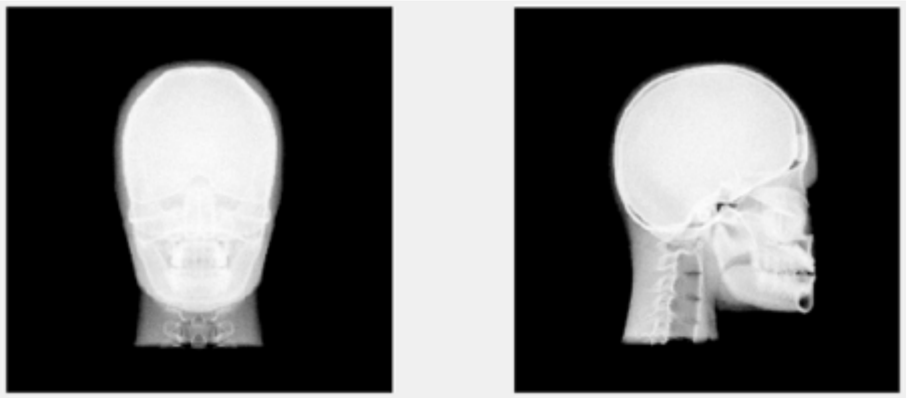


Figure 5.7: FASH head X-ray projections with a 120kVp source with tungsten anode and 4mm aluminium filtration

5.2 Simulation acceleration

Running large simulations in GATE comes at a very high computational cost, so it takes a very long time for them to complete. It is therefore necessary to make adjustments to the simulation setup in order to allow them to run more quickly - ideally a simulation should take no longer than 2-3 days to run for this data collection method to be viable for collecting large amounts of data. There are several modifications that can be made which can speed up the simulations that have been explored in this project.

5.2.1 GPU acceleration

GPU acceleration of simulations would be a very useful feature for these simulations, as it would allow calculations to do with particle interactions during the simulation to be computed in parallel.

In older versions of GATE and GEANT4, GPU acceleration was supported to a certain extent, so in an attempt to utilise these features, I was given access to the university's GPU server, Shannon. After several attempts to build GPU modules into different versions GATE and GEANT4 packages, this experiment ultimately provided a solution that was too temperamental to consider using for simulations with complex geometry, as simulations would crash and produce unexpected results for every simulation that was run with GPU features enabled.

The use of a GPU to run GATE simulations is, however, an interesting feature that may be useful to look into if this project is built upon further in the future.

5.2.2 VRT

GATE can implement a Variance Reduction Technique (VRT) to make the simulation run faster. This works by 'killing' the generated particles that hit the surface of the detector and then computing the mean free path (MFP) of the particle through the detector. The computation of the particle's path may then be performed K times for each particle to avoid generation and propagation of the particle. As K increases, however, the variance of the output data is reduced, so it is recommended that $K \leq 10$. The simulation time should decrease linearly with K.

$$PATH = MFP \times -log(1 - R) \tag{5.1}$$

R is a uniformly distributed random number between 0 and 1.

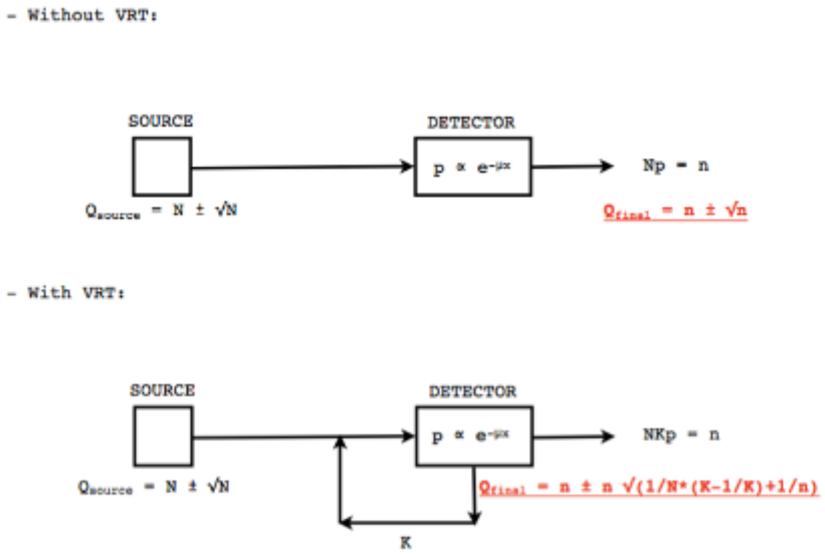


Figure 5.8: This diagram shows the differences between running a simulation with and without VRT [20]

The above figure describes this technique graphically, where N is the mean of the generated particles, and $\sqrt{N}$ is the standard deviation. The binomial probability of the particle being detected is given by p, the mean of the number of detected particles is $n = Np$ and $\sqrt{n}$ is its standard deviation.

5.2.3 Using cluster tools

GATE simulations can be split into several macro files using the 'cluster tools' job splitter feature in GATE. Simulations can be run using the program, Condor, which submits these jobs to a cluster of CPUs at the university. The simulation run time should therefore decrease linearly with the number of

CPUs used. For the simulations using the FASH phantom, I split the simulation into 90 separate jobs, so that each CPU runs the simulation for 2 seconds of the simulation. Each of these macro files will take about 2 days to run the simulation through, but this is a huge improvement, as without splitting the macros it would take weeks to run the same simulation on a single CPU. The best method currently available that I have found is to use a combination of VRT and cluster tools, setting the VRT factor to 10 as recommended in the GATE users guide.

```
gjs -numberofsplits 90 -clusterplatform condor -condorscript ${HOME}/condor.script ct.mac

condor_submit ct.submit
```

Listing 5.1: Script used to split the main GATE CT macro into 90 different jobs for submitting to condor

5.3 Results

5.3.1 CT images

After running the complete CT simulation across 90 CPUs, images were reconstructed in MATLAB, that distinctly show a variety of organs in the slice of interest, as well as some of the expected CT image artefacts. The simulation took about 48 hours to run.

The sinogram is shown in Figure 5.9 and the reconstructed CT images are shown in Figure 5.10. It can be observed that the iterative reconstruction produces the clearest image whereas the simple back-projection produces a very poor quality image, which is too blurry to be used to identify the structure of the organs contained within.

Beam hardening artefacts are the dominating form of image distortion in these CT images. The bright hue around the circumference of the CT slice is due to beam hardening or more specifically, cupping artefact - this artefact is still apparent despite the bow-tie filter's best efforts to correct it. It would require a very carefully designed bow-tie filter with a more complex geometry to correct this type of artefact fully. Another obvious artefact in the CT images is one that is caused by photon starvation in the area in-front of the spine, which contains dark streaking, obscuring the outline of the pancreas. This photon starvation artefact is a result of the spine bone having a much higher density than the area of the phantom surrounding it in this particular CT slice. Additional CT images for the head of the FASH phantom can be found in Appendix C.

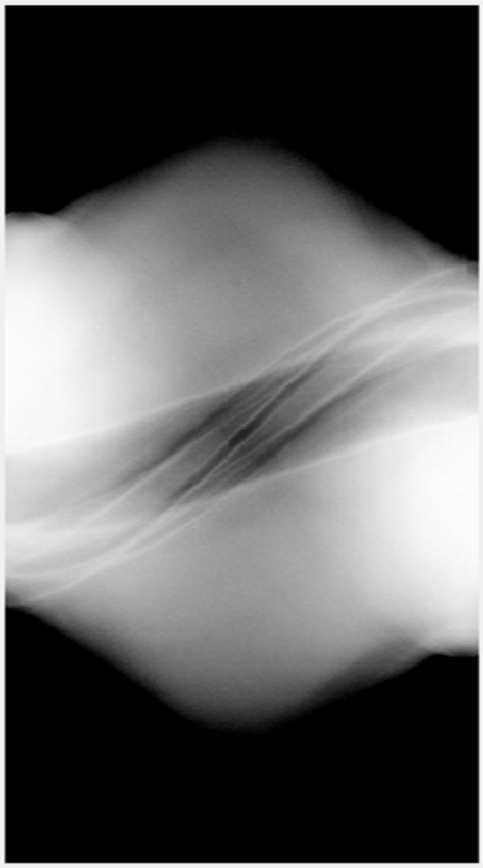


Figure 5.9: Sinogram for simulation with 2 billion primaries and 120kVp source

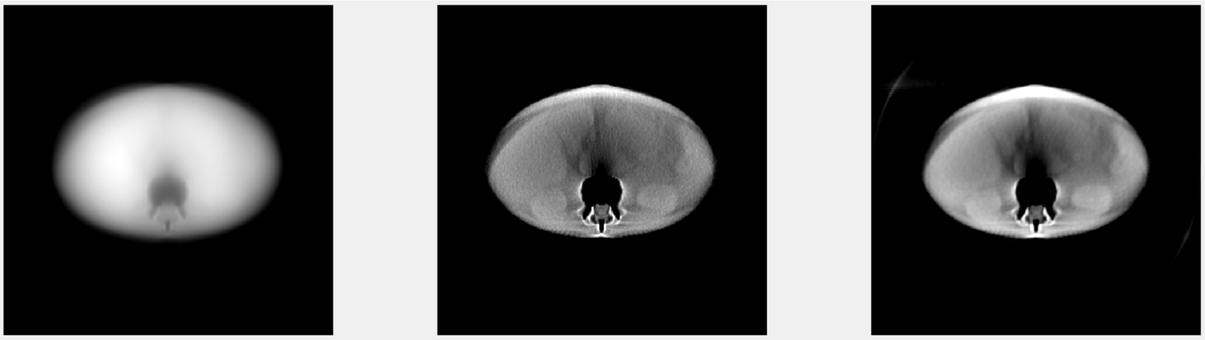
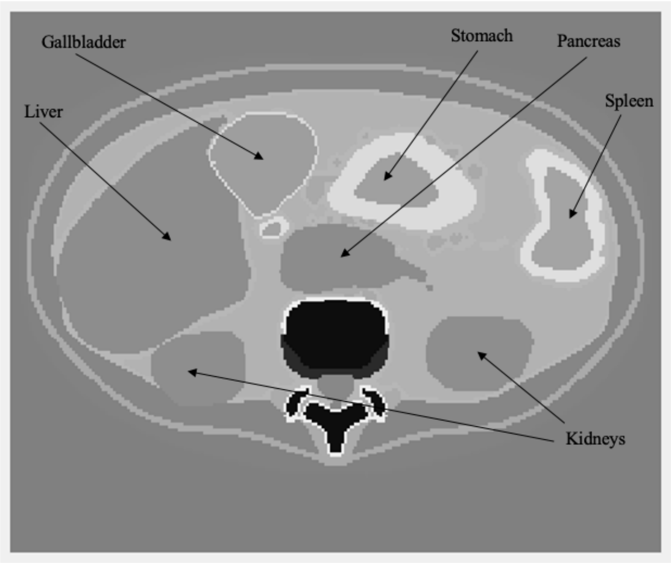
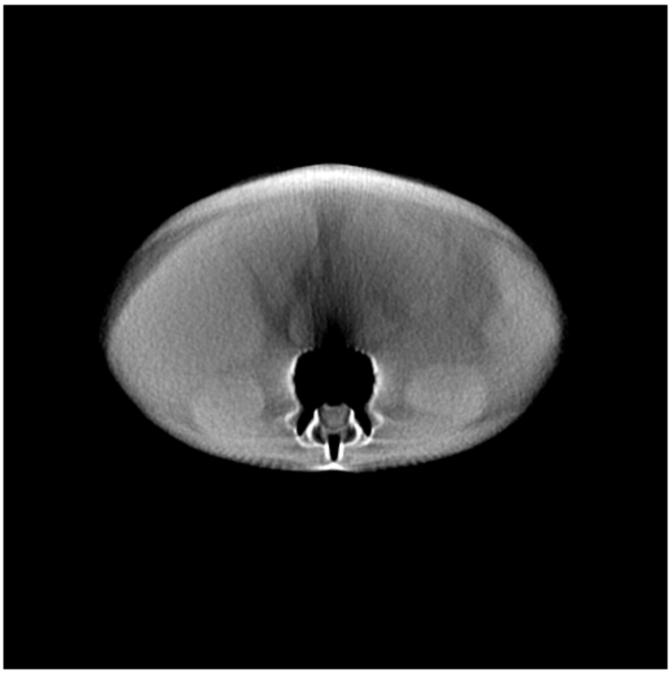


Figure 5.10: For simulation with 2 billion primaries and 120kVp source: a) Back projection b) Filtered back projection c) Iterative reconstruction



(a) Labelled reference image of axial CT slice



(b) Filtered back-projection axial CT slice

Figure 5.11: Comparison of FBP image with labelled reference image for scanned CT slice

Chapter 6

Conclusion and future work

6.1 Conclusion

In this thesis, a framework for simulating a fan-beam CT scanner was developed using GATE. This framework allows the collection of medical images with realistic artefacts. This project has been a success in that it has produced medical images which contain realistic artefacts that can be used in future projects to train machine learning algorithms to assist with image enhancement and organ outlining.

In this project, I have demonstrated that the human phantom (FASH) can be inserted into GATE as a DICOM file for implementation in x-ray and CT simulations. In addition, I have written script in MATLAB that allows the 'ImageCT' output file to be read and manipulated to produce x-ray projections, sinograms and reconstructed CT images by the methods of back-projection, filtered back-projection and iterative reconstruction.

Inspired by the excessively long simulation run times, I explored different options for simulation acceleration, managing to get simulation run times down from 2 weeks to 2 days for large simulations involving 2 billion primary particles through the use of a cluster of CPUs and VRT.

Having been successful in producing CT images, future projects within the university can now build upon this work by refining the scanner geometry and using the collected images for training machine learning algorithms.

6.2 Future work

As this project fits into a system of several projects, it is expected that the work presented in this thesis will be built upon in several different ways towards the goal of training machine learning algorithms by supplying them with a limitless supply of medical imaging data.

If I had more time to work on this project, I would have liked to continue to improve on the FBCT simulation geometry. For example, in my simulations I use a flat detector instead of a curved one so that improvement would have been my next objective. The bow-tie filter geometry can also be refined as, for this project, the filter geometry that I use is quite basic - with more experience using GATE, I think that these changes could become more easily implemented.

As well as CT scanners, GATE allows other types of imaging modules to be simulated. Therefore, it will be possible to use the FASH/MASH human phantoms to collect medical imaging data for different scanner types and geometries to similarly train machine learning algorithms in the future.

In this project, I began to experiment with the use of a GPU to decrease the run time for simulations. Although this feature ultimately proved to be too temperamental for use in my project, in future iterations of this project, it would be worthwhile to explore this feature further, as the university has the necessary resources available.

Chapter 7

Impact and Exploitation

This project facilitates one part in the process of developing a machine learning algorithm for performing CT image enhancement and aiding physicians in the outlining of structures which appear in a CT image. The role of this project is to construct a method for mass medical imaging data collection through the use of the physics simulation program, GATE. This stage of the project is pivotal due to the fact that medical imaging data can not be freely collected by scanning real patients due to the ethical concerns associated with exposing people unnecessarily to dangerous, ionising radiation for data collection purposes.

The initial 'engineering idea' stage to the beginning of the 'further development' stage of the typical development chain for a commercial product as shown in Figure 7.1 has been covered by this project. The 'engineering idea' that has inspired this project is the idea that - 'It might be possible to change the way that physicians interpret medical imaging data through the use of machine learning algorithms'. Taking that idea forward, I have carried out initial research into one aspect of this engineering idea, which is the issue of mass data collection. From the initial research stage, I concluded that it was a viable option to obtain medical imaging data through the use of computer simulations, so proceeded to develop this idea up until the point of achieving my initial goal which was to find a way of producing vast quantities of medical data without the expense and ethical concerns associated with collecting data through real-life scanning procedures.

The next stage in the development of the machine learning algorithms would be to use the script produced in this project to collect large quantities of CT images. The phantom can also be changed in order to collect as much of a variety of data as possible. These images will then be used as part of a future projects within the university to train machine learning algorithms. These algorithms must first be researched and developed by other students before being trained by the CT data. Once this stage of the process is complete, the algorithms should be tested using real life CT images to observe their effectiveness. If this is proven to be effective, the software would then be made commercially available for integration into clinical environments.

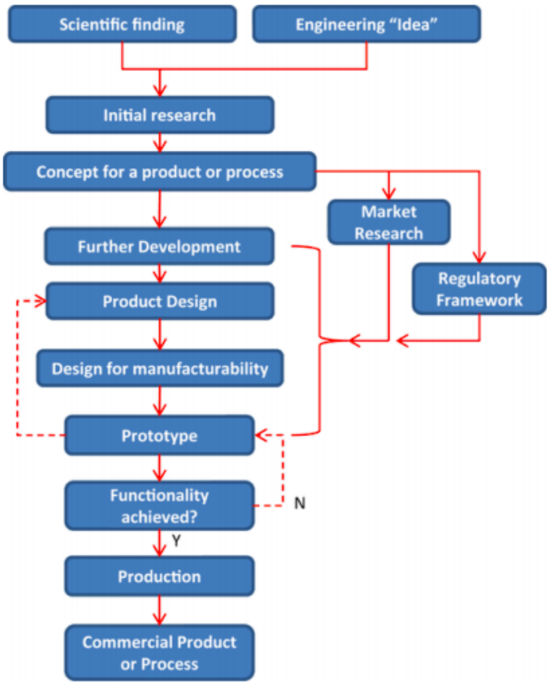


Figure 7.1: Typical product development cycle

If the overarching goal of training machine learning algorithms to interpret medical images is achieved, this would be of enormous benefit to the healthcare industry. The efficiency of the CT scanning procedure could be increased and the human error responsible for mistakes made by physicians in outlining the scans would be reduced. This is in the best interest of patients too as the likelihood of a correct diagnosis from a CT scan would be increased. By streamlining this process, treatment can be delivered as efficiently as possible which will give patients a better outlook.

With fewer mistakes/oversights being made in the interpretation of CT images, money and time can be saved since it wont be as likely that more diagnostic tests will be required later down the line if certain medical conditions are not identified in the initial CT scan. This will therefore be an attractive product for healthcare providers who have to carefully prioritise where budgets are to be spent.

New jobs would be created for engineers who would have to be involved in setting up and maintaining these algorithms in hospitals as well as training radiologists how to operate them in a way which would enhance their diagnostic capabilities. The healthcare industry is always going to be incredibly welcoming of new technology that has the potential to improve the way that its services are delivered to the public, so this project will undoubtedly achieve commercial success if it manages to perform as anticipated.

Acknowledgements

I would like to sincerely thank Dr Dave Laurenson for his help with this project - his enthusiasm for the subject has been contagious. It has been a pleasure to carry out this project under his supervision. Under the circumstances brought on by the COVID-19 pandemic, it has not been an easy year for students completing their theses, but through his resourcefulness, Dr Laurenson has managed to make our interactions very efficient and useful. I would also like to thank my friends and family for their moral support and encouragement throughout this past academic year.

References

[1] Jan, S., Santin, G., Strul, D. et al.: 'GATE: a simulation toolkit for PET and SPECT', in: *Physics in Medicine and Biology* 49.19 (2004), pp. 4543–4561, DOI: 10.1088/0031-9155/49/19/007.

[2] Brun, R. and Rademakers, F.: 'ROOT — An object oriented data analysis framework', in: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 389.1-2 (1997), pp. 81–86, DOI: 10.1016/s0168-9002(97)00048-x.

[3] Abdulla, S.: *Production of X-rays*, Aug. 2020, URL: https://www.radiologycafe.com/radiology-trainees/frcr-physics-notes/production-of-x-rays.

[4] Maier, A.: 'Medical imaging systems: an introductory guide', (Springer Open, 2018).

[5] Lumen: *X Rays: Atomic Origins and Applications*, URL: https://courses.lumenlearning.com/physics/chapter/30-4-x-rays-atomic-origins-and-applications/.

[6] Seibert, J. A. and Boon, J. M.: 'X-ray imaging physics for nuclear medicine technologists. Part 2: X-ray interactions and image formation', in: *Journal of nuclear medicine technology* (Mar. 2005), pp. 3–18.

[7] Hoppe, R., Phillips, T. L., Roach, M. and Leibel, S. A.: 'Leibel and Phillips textbook of radiation oncology', (Elsevier Saunders, 2010).

[8] Price, S.: *Coherent scattering: Radiology Reference Article*, URL: https://radiopaedia.org/articles/coherent-scattering?lang=gb.

[9] Kalender, W. A.: 'X-ray computed tomography', in: 51.13 (June 2006), R29–R43, DOI: 10.1088/0031-9155/51/13/r03, URL: https://doi.org/10.1088/0031-9155/51/13/r03.

[10] Abdulla, S.: *CT equipment*, URL: https://www.radiologycafe.com/radiology-trainees/frcr-physics-notes/ct-equipment.

[11] Wolbarst, A. B., Capasso, P. and Wyant, A. R.: 'Medical imaging: essentials for physicians', (Wiley/Blackwell, 2013).

[12] Fessler, J.: 'Fundamentals of CT Reconstruction in 2D and 3D', in: pp. 263–295, ISBN: 9780444536334, DOI: 10.1016/B978-0-444-53632-7.00212-4.

[13] Greenway, K.: *Hounsfield unit: Radiology Reference Article*, URL: https://radiopaedia.org/articles/hounsfield-unit?lang=gb.

[14] Kissane, J., Neutze, J. A. and Singh, H.: 'Computed Tomography', in: *Radiology Fundamentals* (2020), pp. 27–31, DOI: 10.1007/978-3-030-22173-7_6.

[15] Willemink, M. J. and Noël, P. B.: 'The evolution of image reconstruction for CT—from filtered back projection to artificial intelligence', in: *European Radiology* 29.5 (2018), pp. 2185–2195, DOI: 10.1007/s00330-018-5810-7.

[16] Stiller, W.: 'Basics of iterative reconstruction methods in computed tomography: A vendor-independent overview', in: *European Journal of Radiology* 109 (2018), pp. 147–154, DOI: 10.1016/j.ejrad.2018.10.025.

[17] Veikutis, V., Budrys, T., Basevicius, A. et al.: *Artifacts in computer tomography imaging: how it can really affect diagnostic image quality and confuse clinical diagnosis?*, Mar. 2015, URL: https://www.jvejournals.com/article/15949.

[18] Cassola, V. F., Melo Lima, V. J. de, Kramer, R. and Khoury, H. J.: 'FASH and MASH: female and male adult human phantoms based on polygon mesh surfaces: I. Development of the anatomy', in: *Physics in Medicine and Biology* 55.1 (2009), pp. 133–162, DOI: 10.1088/0031-9155/55/1/009.

[19] Valentin, J.: 'Basic anatomical and physiological data for use in radiological protection: reference values', in: *Annals of the ICRP* 32.3-4 (2002), pp. 1–277, DOI: 10.1016/s0146-6453(03)00002-2.

[20] *GATE Users Guide ¶*, URL: https://opengate.readthedocs.io/en/latest/index.html.

[21] Mooney, C. Z.: 'Monte Carlo simulation', (Sage, 2003).

[22] Sokolmarek: *sokolmarek/CTBasicReconstruction*, URL: https://github.com/sokolmarek/CTBasicReconstruction/blob/main/cART.m.

[23] Wang, J. and Fleischmann, D.: 'Improving Spatial Resolution at CT: Development, Benefits, and Pitfalls', in: *Radiology* 289.1 (2018), pp. 261–262, DOI: 10.1148/radiol.2018181156.

[24] White, D. R., Woodard, H. Q. and Hammond, S. M.: 'Average soft-tissue and bone models for use in radiation dosimetry', in: *The British Journal of Radiology* 60.717 (1987), pp. 907–913, DOI: 10.1259/0007-1285-60-717-907.

[25] Poludniowski, G., Landry, G., DeBlois, F., Evans, P. M. and Verhaegen, F.: 'SpekCalc: a program to calculate photon spectra from tungsten anode x-ray tubes', in: *Physics in Medicine and Biology* 54.19 (2009), DOI: 10.1088/0031-9155/54/19/n01.

[26] Mason, J. H.: 'Quantitative Cone-Beam Computed Tomography Reconstruction for Radiotherapy Planning', in: (Oct. 2018).

Appendix A

MATLAB code

All code used to produce the final images has been uploaded to the following GitHub repository: https://github.com/Aah66/GATE_CT

A.0.1 CT cluster MATLAB code

```
%this program reads in FBCT projections in the form of .dat files and
%produces a sinogram. From that, a back projection, filtered back
%projection, and an iterative reconstruction are produced.

%projection set (1-4)
projection_set=1;

%number of projection files/angles
numfiles =180;

% number of pixels in detector
LENGTH=512;

%number of splits in cluster
splits=90;

% Set image contrast (must be between 0 and 1, eg. 0.25)
contrast=0.15;

% Set number of iterations for iterative reconstruction (eg. 50)
iterations=1;




%the following is a quick way of reading in files ct_projection_000.dat to
%ct_projection_180.dat
i=0;
```

```
image=0;

for k = 0:9
if (k>(numfiles*i/splits)-2)
    i=i+1;
end


  file = sprintf('../projections%d/ct_projection%d_00%d.dat', projection_set, i, k);
  fp = fopen(file,'rb');
data = fread(fp,LENGTH*1*1,'float32');
image(1:LENGTH,k+1)=[data];
fclose(fp);


end

for k = 10:99

if (k>(numfiles*i/splits)-2)
    i=i+1;
end

file = sprintf('../projections%d/ct_projection%d_0%d.dat', projection_set, i, k);
fp = fopen(file,'rb');
data = fread(fp,LENGTH*1*1,'float32');
image(1:LENGTH,k+1)=[data];
fclose(fp);

end

for k = 100:numfiles-2

if (k>(numfiles*i/splits)-2)
    i=i+1;
end

file = sprintf('../projections%d/ct_projection%d_%d.dat', projection_set, i, k);
fp = fopen(file,'rb');
data = fread(fp,LENGTH*1*1,'float32');
image(1:LENGTH,k+1)=[data];

fclose(fp);

end

for k = numfiles-1

if (k>(numfiles*i/splits)-1)
    i=i+1;
```

```
end

file = sprintf('../projections%d/ct_projection%d_%d.dat', projection_set , i, k);
fp = fopen(file,'rb');
data = fread(fp,LENGTH*1*1,'float32');
image(1:LENGTH,k+1)=[data];

fclose(fp);

end

% rescale max and min values of sinogram to between 0 and 1
image = image - min(min(abs(image)));
image = image/max(max(abs(image)));


% adjust contrast to optimise visibility
image = imadjust(image, [0 contrast], [0 1]);

%invert sinogram
inverted=imcomplement(image);


% performs unfiltered back projection
I1 = iradon(inverted,0:numfiles-1,'linear','none',LENGTH);
I1 = I1 - min(min(abs(I1)));
I1 = I1/max(max(abs(I1)));
I1=imrotate(I1, 270);
I1=flip(I1,2);

%performs filtered back projection
I2 = iradon(inverted,0:numfiles-1,'linear','hann', LENGTH);
I2 = I2 - min(min(abs(I2)));
I2 = I2/max(max(abs(I2)));
I2=imrotate(I2, 270);
I2=flip(I2,2);


%performs iterative reconstruction
I3 = cART(inverted,numfiles, iterations);
I3 = I3 - min(min(abs(I3)));
I3 = I3/max(max(abs(I3)));
I3=imrotate(I3,180);


%shows unfiltered back projection image
figure();
subplot(131)
imshow(I1(:,:,1),[0.5 1]);
```

```
%shows filtered back projection image
subplot(132)
imshow(I2(:,:,1),[0.1 0.5]);

%shows iterative back projection image
subplot(133)
imshow(I3(:,:,1),[0.1 0.65]);



%shows inverted sinogram image
figure();
imshow(inverted(:,:,1),[]);

% %shows non-inverted sinogram image
% figure();
% imshow(image(:,:,1),[]);
```

Appendix B

GATE code

B.0.1 Main CT macro

```
/control/execute mac/verbose.mac

#=====================================================
# GEOMETRY
#=====================================================

/control/execute mac/geometry.mac

# Phantom
/control/execute mac/fash.mac
#=====================================================
# PHYSICS
#=====================================================

/control/execute mac/physics.mac

#=====================================================
# SIMULATION STATISTICS
#=====================================================

#/gate/actor/addActor   SimulationStatisticActor stat
#/gate/actor/stat/save  output/stat-ct.txt

#=====================================================
# INITIALISATION
#=====================================================

/gate/run/initialize

#=====================================================
# DIGITIZER
#=====================================================
/gate/digitizer/Singles/insert adder
/gate/digitizer/Singles/insert readout
/gate/digitizer/Singles/readout/setDepth 2
/gate/digitizer/Singles/insert thresholder
/gate/digitizer/Singles/thresholder/setThreshold 10 keV

#=====================================================
# SOURCE
#=====================================================

/control/execute mac/source2.mac


#=====================================================
# VISUALISATION
#=====================================================

#/control/execute mac/viewer.mac

#=====================================================
# OUTPUTS
#=====================================================

/control/execute mac/output.mac

#=====================================================
# START ACQUISITION
#=====================================================

/control/execute mac/acquisition.mac
```

B.0.2 Acquisition

```
# JamesRandom Ranlux64 MersenneTwister
/gate/random/setEngineName MersenneTwister
/gate/random/setEngineSeed auto


################################
# ACQUISITION for 180 projections #
################################
/gate/application/setTimeSlice     1. s
/gate/application/setTimeStart     0.  s
/gate/application/setTimeStop      180. s

/gate/application/setNumberOfPrimariesPerRun 2000000000
/gate/application/start
```

B.0.3 Physics

```
/gate/physics/addProcess PhotoElectric
/gate/physics/processes/PhotoElectric/setModel StandardModel

/gate/physics/addProcess Compton
/gate/physics/processes/Compton/setModel StandardModel

/gate/physics/addProcess RayleighScattering
/gate/physics/processes/RayleighScattering/setModel PenelopeModel

/gate/physics/addProcess ElectronIonisation
/gate/physics/processes/ElectronIonisation/setModel StandardModel e-
/gate/physics/processes/ElectronIonisation/setModel StandardModel e+

/gate/physics/addProcess Bremsstrahlung
/gate/physics/processes/Bremsstrahlung/setModel StandardModel e-
/gate/physics/processes/Bremsstrahlung/setModel StandardModel e+

/gate/physics/addProcess eMultipleScattering e+
/gate/physics/addProcess eMultipleScattering e-


/gate/physics/processList Enabled
/gate/physics/processList Initialized
```

B.0.4 Geometry

```
/gate/geometry/setMaterialDatabase /home/s1606384/GATE-build/Gate-8.2/FBCT_phantom/data/
    MyMaterialDatabase.db

# World
/gate/world/geometry/setXLength 1000. mm
/gate/world/geometry/setYLength 1000. mm
/gate/world/geometry/setZLength 1500. mm
/gate/world/setMaterial Air
/gate/world/vis/forceWireframe
/gate/world/vis/setColor black

#Bowtie filter
/gate/world/daughters/name                wedge0
/gate/world/daughters/insert              wedge
/gate/wedge0/geometry/setXLength           100 mm
/gate/wedge0/geometry/setNarrowerXLength   1. mm
/gate/wedge0/geometry/setYLength           40. mm
/gate/wedge0/geometry/setZLength           200. mm
/gate/wedge0/setMaterial                   Aluminium
/gate/wedge0/vis/forceSolid
/gate/wedge0/vis/setColor                  yellow
/gate/wedge0/placement/setTranslation -75. 0. -550. mm
```

```
/gate/wedge0/placement/setRotationAxis 1 0 0
/gate/wedge0/placement/setRotationAngle 90. deg

/gate/world/daughters/name                wedge3
/gate/world/daughters/insert              wedge
/gate/wedge3/geometry/setXLength           100 mm
/gate/wedge3/geometry/setNarrowerXLength   1. mm
/gate/wedge3/geometry/setYLength           40. mm
/gate/wedge3/geometry/setZLength           200. mm
/gate/wedge3/setMaterial                   Aluminium
/gate/wedge3/vis/forceSolid
/gate/wedge3/vis/setColor                  yellow
/gate/wedge3/placement/setTranslation -75. 0. -590.0 mm
/gate/wedge3/placement/setRotationAxis 1 0 0
/gate/wedge3/placement/setRotationAngle -90. deg

/gate/world/daughters/name                wedge1
/gate/world/daughters/insert              wedge
/gate/wedge1/geometry/setXLength           100 mm
/gate/wedge1/geometry/setNarrowerXLength   1. mm
/gate/wedge1/geometry/setYLength           40. mm
/gate/wedge1/geometry/setZLength           200. mm
/gate/wedge1/setMaterial                   Aluminium
/gate/wedge1/vis/forceSolid
/gate/wedge1/vis/setColor                  yellow
/gate/wedge1/placement/setTranslation 75. 0. -550. mm
/gate/wedge1/placement/setRotationAxis 0 1 1
/gate/wedge1/placement/setRotationAngle 180. deg

/gate/world/daughters/name                wedge4
/gate/world/daughters/insert              wedge
/gate/wedge4/geometry/setXLength           100 mm
/gate/wedge4/geometry/setNarrowerXLength   1. mm
/gate/wedge4/geometry/setYLength           40. mm
/gate/wedge4/geometry/setZLength           200. mm
/gate/wedge4/setMaterial                   Aluminium
/gate/wedge4/vis/forceSolid
/gate/wedge4/vis/setColor                  yellow
/gate/wedge4/placement/setTranslation 75. 0. -590. mm
/gate/wedge4/placement/setRotationAxis 0 -1 1
/gate/wedge4/placement/setRotationAngle 180. deg

#Collimator 1
/gate/world/daughters/name collimator1
/gate/world/daughters/insert box
/gate/collimator1/geometry/setXLength  60. cm
/gate/collimator1/geometry/setYLength  30.0 cm
/gate/collimator1/geometry/setZLength  5. cm
/gate/collimator1/placement/setTranslation 0. 15.1 -60. cm
/gate/collimator1/setMaterial Lead
```

```
/gate/collimator1/vis/forceWireframe
/gate/collimator1/vis/setColor blue

/gate/world/daughters/name collimator2
/gate/world/daughters/insert box
/gate/collimator2/geometry/setXLength  60. cm
/gate/collimator2/geometry/setYLength  30.0 cm
/gate/collimator2/geometry/setZLength  5. cm
/gate/collimator2/placement/setTranslation 0. -15.1 -60. cm
/gate/collimator2/setMaterial Lead
/gate/collimator2/vis/forceWireframe
/gate/collimator2/vis/setColor blue

#Collimator 2
/gate/world/daughters/name collimator3
/gate/world/daughters/insert box
/gate/collimator3/geometry/setXLength  60. cm
/gate/collimator3/geometry/setYLength  30.0 cm
/gate/collimator3/geometry/setZLength  5. cm
/gate/collimator3/placement/setTranslation 0. 15.1 -50. cm
/gate/collimator3/setMaterial Lead
/gate/collimator3/vis/forceWireframe
/gate/collimator3/vis/setColor blue

/gate/world/daughters/name collimator4
/gate/world/daughters/insert box
/gate/collimator4/geometry/setXLength  60. cm
/gate/collimator4/geometry/setYLength  30.0 cm
/gate/collimator4/geometry/setZLength  5. cm
/gate/collimator4/placement/setTranslation 0. -15.1 -50. cm
/gate/collimator4/setMaterial Lead
/gate/collimator4/vis/forceWireframe
/gate/collimator4/vis/setColor blue


# CTscanner system
/gate/world/daughters/name CTscanner
/gate/world/daughters/insert box
/gate/CTscanner/geometry/setXLength  512. mm
/gate/CTscanner/geometry/setYLength  10. mm
/gate/CTscanner/geometry/setZLength   10. mm
/gate/CTscanner/placement/setTranslation 0. 0. 45. cm
/gate/CTscanner/setMaterial Air
/gate/CTscanner/vis/forceWireframe
/gate/CTscanner/vis/setColor white

# Module
/gate/CTscanner/daughters/name module
/gate/CTscanner/daughters/insert box
/gate/module/geometry/setXLength  512. mm
```

```
/gate/module/geometry/setYLength  10. mm
/gate/module/geometry/setZLength   10. mm
/gate/module/setMaterial Air
/gate/module/vis/forceWireframe
/gate/module/vis/setColor white

# Cluster
/gate/module/daughters/name cluster
/gate/module/daughters/insert box
/gate/cluster/geometry/setXLength 512. mm
/gate/cluster/geometry/setYLength   10. mm
/gate/cluster/geometry/setZLength   10. mm
/gate/cluster/setMaterial Air
/gate/cluster/vis/forceWireframe
/gate/cluster/vis/setColor white

# Pixel
/gate/cluster/daughters/name pixel
/gate/cluster/daughters/insert box
/gate/pixel/geometry/setXLength 1.  mm
/gate/pixel/geometry/setYLength 10. mm
/gate/pixel/geometry/setZLength 10. mm
/gate/pixel/setMaterial Silicon
/gate/pixel/vis/forceSolid
/gate/pixel/vis/setColor red
/gate/pixel/repeaters/insert cubicArray
/gate/pixel/cubicArray/setRepeatNumberX 512
/gate/pixel/cubicArray/setRepeatNumberY 1
/gate/pixel/cubicArray/setRepeatNumberZ  1
/gate/pixel/cubicArray/setRepeatVector 1. 1. 0. mm

/gate/systems/CTscanner/module/attach module
/gate/systems/CTscanner/cluster_0/attach cluster
/gate/systems/CTscanner/pixel_0/attach pixel

/gate/pixel/attachCrystalSD
```

B.0.5 Source

```
# Xray source (defined using a histogram to define the energy distribution)

/gate/source/verbose 0
/gate/source/addSource histogram gps
/gate/source/histogram/gps/type Plane
/gate/source/histogram/gps/shape Rectangle
/gate/source/histogram/gps/halfx 0.025 mm
/gate/source/histogram/gps/halfy 0.025 mm
/gate/source/histogram/gps/centre 0. 0. -640. mm
/gate/source/histogram/gps/mintheta  0. deg
/gate/source/histogram/gps/maxtheta 14.2 deg
```

```
/gate/source/histogram/gps/angtype iso
/gate/source/histogram/gps/particle gamma
/gate/source/histogram/gps/energytype UserSpectrum
/gate/source/histogram/gps/setSpectrumFile mac/120keV.txt
/gate/source/list
```

Appendix C

Additional Images

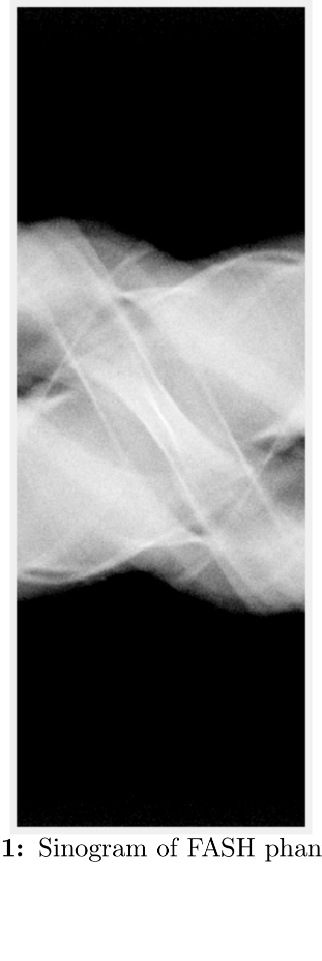
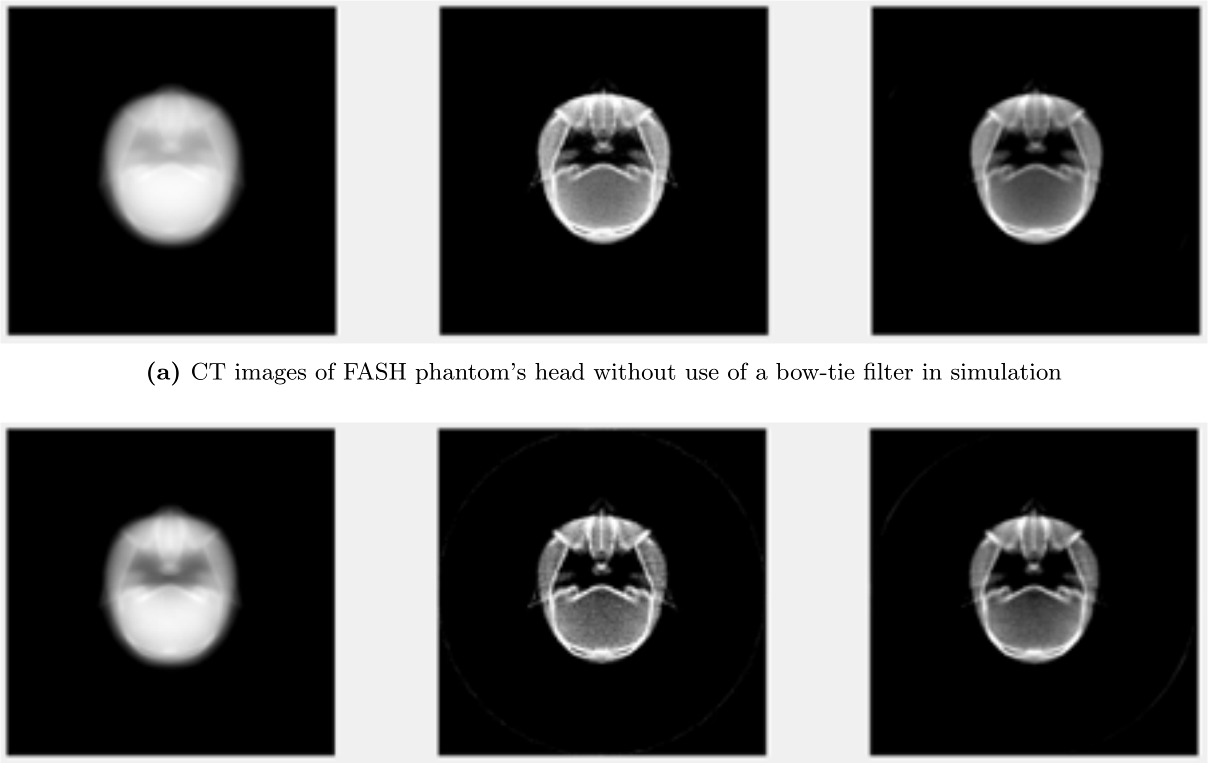


Figure C.1: Sinogram of FASH phantom's head



(a) CT images of FASH phantom's head without use of a bow-tie filter in simulation

(b) CT images of FASH phantom's head with a bow-tie filter included in the simulation geometry

Figure C.2: Comparison of FASH phantom head CT scans with and without the inclusion of a bow-tie filter in the simulation geometry