

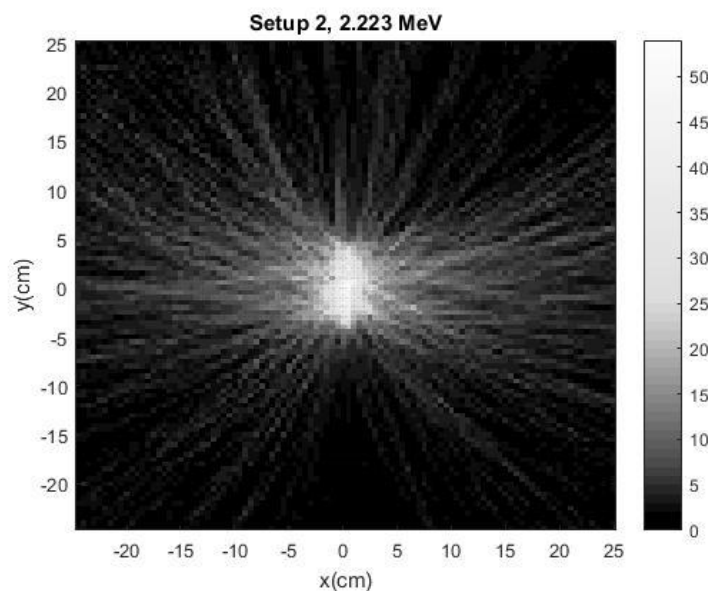


university of
 groningen

kvi - center for advanced
 radiation technology

Bachelor research project Physics - 2017-2018

On the optimal orientation of a thin cylindrical germanium detector for Compton imaging of high-energy gamma rays.



Leon de Jong

May 2018

First examiner: Dr. Peter Dendooven

Second examiner: Dr. Emiel van der Graaf

Table of Contents

1	Introduction	3
2	Theory	4-6
2.1	Photon interactions	4-6
2.1.1	Photoelectric effect	4
2.1.2	Pair Production	4-5
2.1.3	Coherent scattering	5
2.1.4	Compton scattering	5-6
2.2	Proton therapy	7-10
2.2.1	Proton therapy vs photon therapy	7-9
2.2.2	Alternatives to protons	9
2.2.3	Dose control	9-10
2.2.4	Energies of interest	10
2.3	Compton Imaging	11-15
2.3.1	Compton Camera	11-12
2.3.2a	Two Stage Compton Camera	12
2.3.2b	Multistage Compton Camera	12
2.3.2c	Single Stage Compton Camera	12
2.3.3	Cones	13-14
2.3.4	Conic Sections	14
2.3.4a	Circle	14
2.3.4b	Ellipse	14-15
2.3.4c	Hyperbola	15
2.3.4d	Parabola	15
2.3.5	Detector Material	15
3	Materials and methods	16-30
3.1	Monte Carlo Simulations	16-17
3.2	Detector size and material	17
3.3	Matlab	17
3.4	Cones and ellipses	18-19
3.5	Grids	20
3.6	Newton-Raphson Method	20-21
3.7	Intersections	21-24
3.8	Tracking	25
3.9	Image Creation	25-27
3.9a	The distance d travelled through a pixel	27-28
3.9b	Matching d with the correct pixel	28-29
3.10	Full Width at Half Maximum	30
4	Results	31-34
5	Discussion	35-36
5.1	Optimal orientation	35
5.2	Code Improvements	35-36
6	Conclusion	37
7	References	38-39

1. Introduction

At the moment, proton therapy is a rapidly growing method for the treatment of cancer. Instead of photons, high-energy protons are employed to battle tumors that arise in human bodies. During one session, multiple beams of these positively-charged particles are fired in the direction of the designated target area. However, a lot of elements can be improved to maximize the efficiency of proton therapy.

A large improvement would be the ability to monitor the energy deposit in human tissue of every beam in real-time and to be able to change the energies of the subsequent beams of protons accordingly. Compton Cameras are a promising piece of technology to fulfill this role. Their function is to catch Prompt Gammas(PGs) that get emitted by the tissue excited due to the irradiation. The collected data can be used to create an image of the location of the excited tissue, showing which location in the body was traversed by the protons. To create a reliable and accurate image, the Compton Camera should be able to catch enough PGs. The challenge lies in the quest to find a Compton Camera with a high enough efficiency to do so. At the moment, a lot of semiconductors and scintillators are being tested to check whether they are suitable for this job. This paper researches the properties of a germanium-based Compton Camera.

The focus of this study lies in comparing two different setups for a Germanium(Ge) detector. The detector is cylindrical, with a diameter of 91 mm and a thickness of 11 mm. The first setup has the side of the cylinder facing the PG source, while the second setup has the detector facing the source with one of its two circular areas.

Another goal for the study is to create a script in the scientific programming tool MATLAB, with the goal to be usable in the future to compare other detectors. To test other detectors, with any other shapes and made of any suitable material, the only data needed to run the script is the data out of Monte Carlo simulations for these detectors.

The main question the paper is trying to answer is:

In which orientation, sideways or frontal, does a thin cylindrical germanium detector perform better for Compton imaging of high-energy gamma rays?

2. Theory

2.1 Photon interactions

Since the focus of the study lies on photons, it is a good start to know the influence matter can have on photons. When a photon travels through matter, there are several processes that can occur that either absorb or scatter the photon.

2.1.1 Photoelectric effect

In the photoelectric effect,, an photon gets completely absorbed by an atom. Subsequently, the atom gets ionized due to the ejection of a photoelectron. A necessity for this effect is that the photon energy has to be higher than the binding energy of one of the electrons of the atom. This can be seen in *figure 1*. The ejected photoelectron will have an energy equal to the energy of the photon minus the binding energy of the electron:

$$E_{\text{photon}} = h\nu$$

$$E_e = h\nu - E_{\text{binding}}$$

This interaction has the highest probability to happen if the binding energy is a fraction smaller than the photon energy, so most of the time the energy of the ejected electron will be small. However, if the photoelectron energy is high enough, the photoelectron can be the cause of the occurrence of another photoelectric effect. This effect is mostly associated with relatively low photon energies and high atomic number Z. The probability of photoelectric absorption, τ , depends on the atomic number and photon energy as follows:

$$\tau \propto \frac{Z^n}{(h\nu)^3} \text{ with } n \text{ ranging from 3 to 4 over the gamma-ray energy spectrum.}$$

τ stands for the probability of photoelectric absorption. This is why gamma-ray shields mostly contain elements with high atomic numbers, like lead ($Z=82$). (How photons interact with matter n.d., Interactions of Photons with Matter The probability of an interaction per g cm⁻² of material traversed. Units of cm² g n.d., PHYS 352 Photon Interactions Photon Interactions in Matter n.d.)

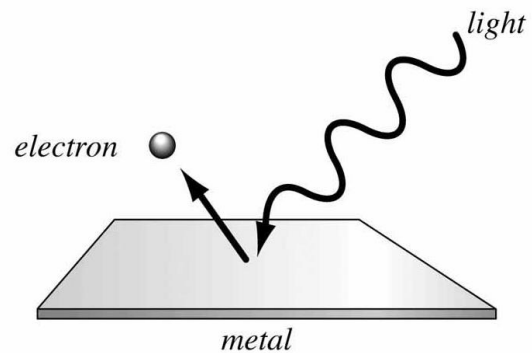


Figure 1: the photoelectric effect (source: http://ecetutorials.com/wp-content/uploads/2014/01/170450_46502_68.jpg)

2.1.2 Pair Production

Another interaction that can occur when a photon travels through matter is called pair production. *Figure 2* shows the interaction. When the photon travels close to the nucleus of an atom or an electron, the photon interacts with the particle, forming an electron and positron pair. This process is driven by the presence of an electromagnetic field. The lower photon energy limit for this

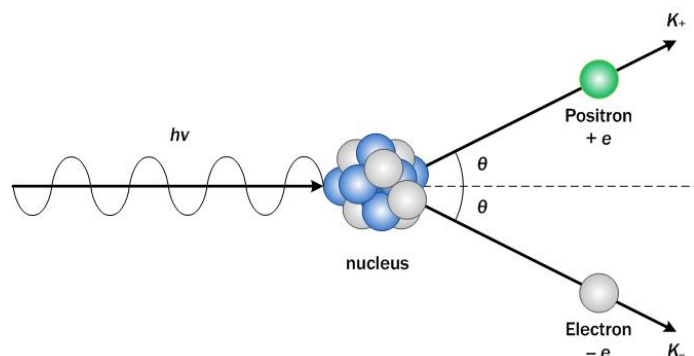


Figure 2: Pair production(<https://www.quora.com/If-antimatter-and-matter-combine-together-energy-is-released-and-no-matter-is-left-Does-this-violate-the-law-of-conservation-of-mass>)

procedure is the rest mass energy of the created pair, but since they both have the same mass, this is equal to two times the rest mass energy of an electron, $2m_e c^2$:

$$E_{\text{photon}} \geq 2m_e c^2 = 1.022 \text{ MeV}$$

The total kinetic energy of the electron and positron pair equals the photon energy minus the photon energy lower limit.

$$E_{\text{electron}} + E_{\text{positron}} = h\nu - 1.022 \text{ MeV}$$

Pair production probability grows with increasing photon energy and with atomic number.

$$\kappa \propto Z^2$$

2.1.3 Coherent scattering

Coherent scattering, also known as elastic scattering, is known as the interaction during which the photon hits a particle and bounces off, leaving the photon energy unchanged. Examples are Thomson scattering and Rayleigh scattering. This can be seen in *figure 3*.

Thomson scattering is the interaction during which a photon collides with a charged particle. Subsequently, because a photon is an electromagnetic wave, a Lorentz force arises and the charged particle is accelerated. This leads to the emission of radiation by the particle at the expense of energy of the initial photon. This radiation comes in the form of a new photon, sent into a random direction. As a simplification this can be seen as the 'scattering' of the incident photon. It is mostly associated with photons with low energies, orders of magnitudes smaller than $m_e c^2$. (Thomson Scattering n.d.)

$$h\nu \ll m_e c^2$$

The effect is frequently seen when dealing with plasma's.

During Rayleigh scattering, photons scatter off of atoms as large as a tenth of the wavelength of the photon. The intensity I of the scattered photon is very dependent of wavelength. T

$$I \propto \frac{1}{\lambda^4}$$

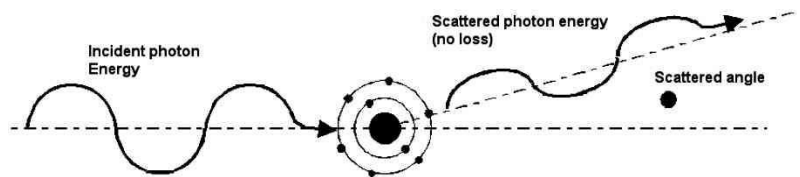


Figure 3: Coherent scattering (source: <https://studentradiographer.com/?p=420>)

Shorter waves get scattered with higher efficiency. Blue photons get scattered more regularly than all other photons in the visible spectrum. The light emitted by the sun contains a spectrum of colors, but the shorter wavelengths get scattered down to earth more efficiently. A consequence is that the sky looks blue for humans. (Blue Sky and Rayleigh Scattering n.d.)

2.1.4 Compton scattering

Compton scattering is an interaction between a high-energy photon and a free electron in matter. This can be seen in *figure 4*. When dealing with bound electrons in matter, the energy of the photon is magnitudes larger than the binding energy of the electron. This leads to the assumption that the binding energy of the electron is insignificant and that the electron can be seen as a free electron. The electromagnetic wave clashes with the electron and the electron and photon scatter.

During the collision, some photon energy goes to the electron. The transferred energy can range from almost nothing to a large amount of the photon energy.

The scatter angle of the photon can be calculated using the following formula:

$$\cos \theta_1 = 1 + m_e c^2 \left(\frac{1}{E_0} - \frac{1}{E_0 - \Delta E_1} \right)$$

In the formula, E_0 is the initial PG energy, ΔE_1 is the energy deposited in the electron by the photon and θ_1 is the Compton angle, which will be discussed in the section Compton Cones.

This scattering is interesting for medical purposes since it is the most seen interaction in soft tissue in the energy range from 100 keV to 10 MeV. Furthermore is the probability of the effect occurring almost independent of atomic number Z and the highest probability can be found in the photon energy range 50-100 keV. It also scales with the number of electrons per gram. (Johns and Cunningham n.d.; Pajevic n.d.; PHYS 352 Photon Interactions Photon Interactions in Matter n.d.)

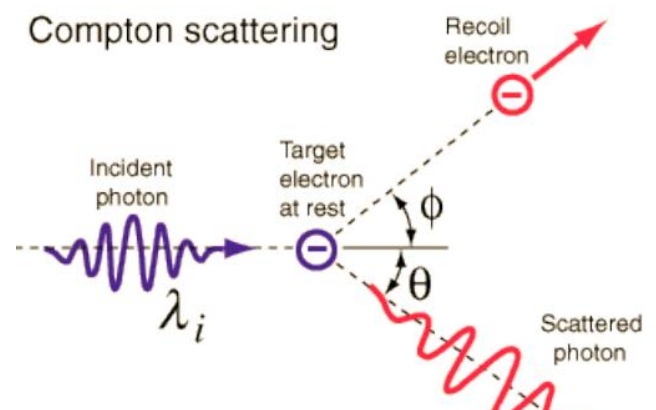


Figure 4: Compton scattering(source: <https://www.physics.queensu.ca/~phys352/lect17.pdf>)

2.2 Proton Therapy

2.2.1 Proton therapy vs photon therapy

Proton therapy offers some advantages over photon therapy. A big benefit of proton therapy over photon therapy is the existence of a 'Bragg Peak' in the case of protons. This can be seen in *figure 5 and 6*. When looking at ionizing radiation, like protons, and plotting the distance travelled through matter, for example, human tissue, versus the energy loss of the radiation, the place with the highest energy loss is called the Bragg peak. The plot is known as the Bragg curve. In case of proton and photon therapy, the traversed matter is human tissue. A beam consisting of the massless and uncharged photons penetrates any quantity of tissue quite easily while steadily depositing energy. Highest energy deposition is found in the first few centimeters, increasing in depth for higher energies, as seen in figure 5.

A downside of photon irradiation is the damage done to tissue surrounding the tumor. Usually, the tumor is found several centimeters under the surface of the skin. (Abril et al. 2013)(Why is proton therapy a preferable option, and what is the Bragg peak? | Proton Therapy Today n.d.) Since the largest dose is delivered in the first few centimeters that get transversed by a photon beam coming from one direction, most damage is done in these regions, and not in the tumor. Also, the regions behind the tumor are affected by the beam of high-energy photons. A way to transport less energy to healthy tissue and a way to shift the Bragg peak more precisely to the location of the tumor would be major improvements in the effective treatment of cancer.

Proton therapy is found to be a step in the right direction. When these high-energy ions enter tissue, they quickly start losing velocity continuously as they dig deeper into the body due to interaction with the human tissue. (Mohan and Grosshans 2016) When looking more precisely at the process, three types of interaction can be distinguished. First of all, there is a Coulomb interaction between protons and atomic electrons. Secondly, the protons undergo Coulomb interactions with nuclei. Thirdly, nuclear interactions occur. Meanwhile, the energy they deliver increases for lower velocities. The Bragg curve is almost flat for this process. For protons, the energy deposition per travelled distance has its peak it almost comes to a stop. This is seen in the Bragg curve as a fast climb of the curve, resulting in a peak. After the peak, the energy deposition plummets to zero almost immediately. This

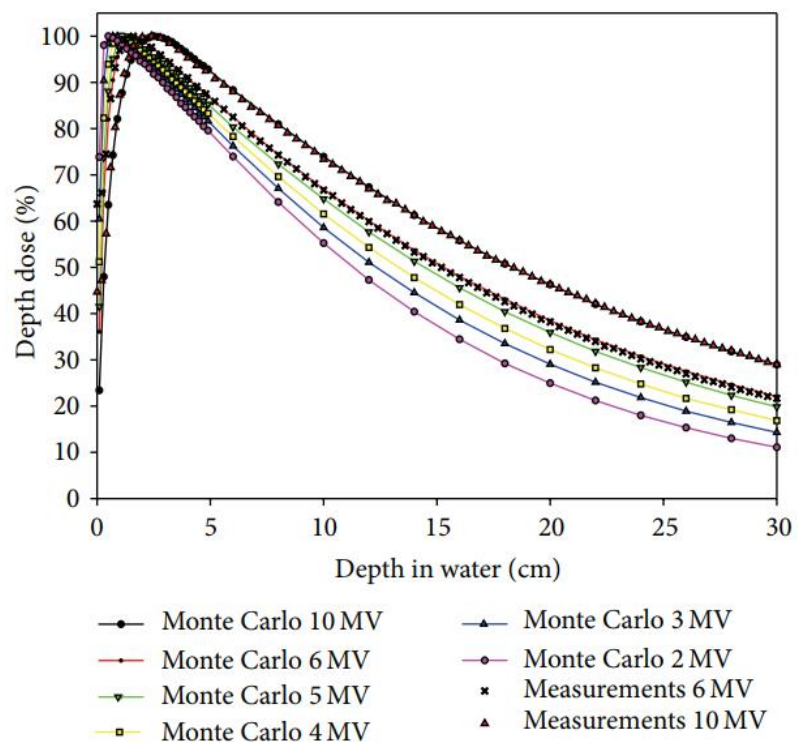


Figure 5: Energy deposition per cm for photons of different energy levels(source: Y. Zhang)

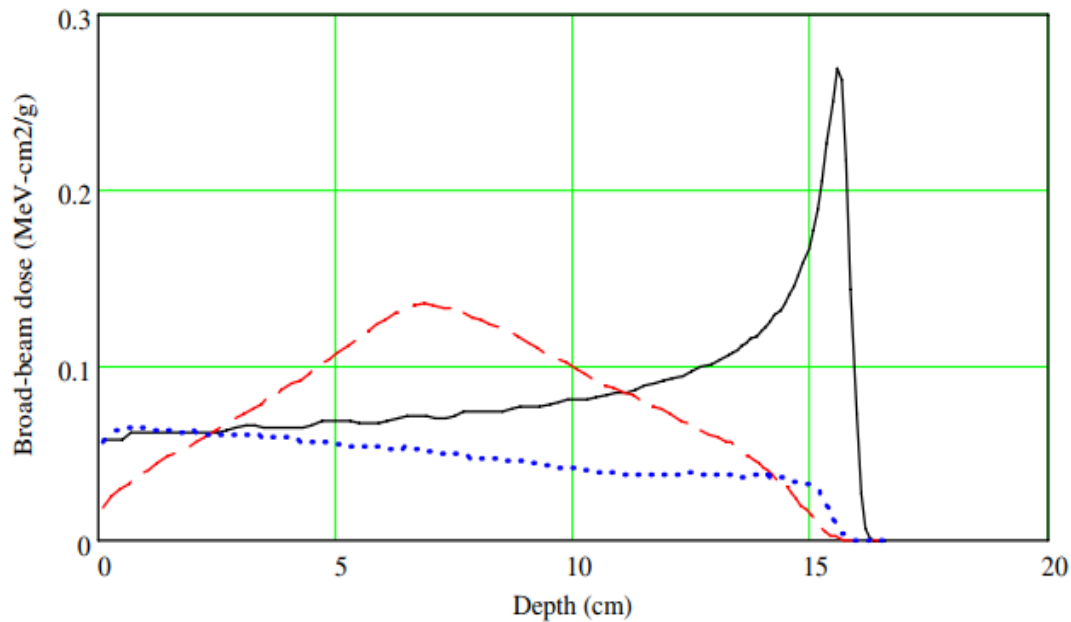


Figure 6: energy deposition per cm for 150 MeV protons(solid), deuterons(dashed) and alpha particles(dashed). Dose multiplied by 0.01(black), (Source: D. Jette)

process can be seen in *figure 6* as the solid line. The most obvious benefit of the proton beam is the possibility to target most of the dose of the particle beam on a specific spot.

However, when using an mono-energetic proton beam, the Bragg peak is too narrow to combat a tumor effectively. A remedy for this problem is to use protons with a range of energies. The peak will become wider using this technique. This is called a spread-out Bragg Peak(SOBP). (Jette and Chen 2011) *Figure 7* shows a comparison of dose distribution in photon and proton therapy.

A formula for the energy dose lost as a charged particle travels through matter, also known as stopping power, was made by the famous nuclear physicist Hans Albrecht Bethe, known as one of the greatest physicists of the twentieth century. He did a lot of great work for astrophysics, quantum electrodynamics and solid-state physics, and he won a Nobel prize for his work on energy production in stars. He is also known for his big contributions to the creation of the hydrogen bomb during World War II. (Lee and Brown 2007; Narins 1995)

The Bethe stopping formula takes the following form:

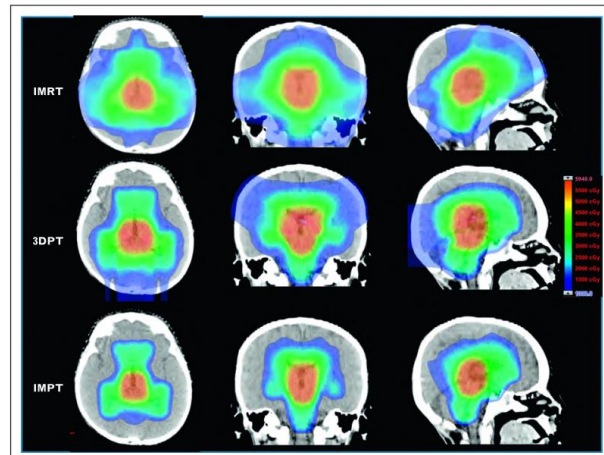


Figure 7: a comparison of dose distribution during radiotherapy(first row) and two different forms of proton therapy(second and third row)(image: E. Merchant)

$$-\frac{dE}{dx} = \frac{4\pi e^4 Z_1^2 Z_2 N}{v^2} \left[\ln \left(\frac{2m_e v^2}{I(1 - (v/c)^2)} \right) - (v/c)^2 \right]$$

With e the electron charge, Z_1 the atomic number of the particle, Z_2 the atomic number of the matter, N the Avogadro number, c the speed of light, m_e the mass of an electron, v the velocity of the particle, and I the mean ionization potential

As discussed earlier, the energy dose delivered by a particle per distance travelled in a medium increases when the velocity of the particle decreases. Looking at the Bethe formula, one can see that this energy deposition per unit length is inversely proportional with v^2 , confirming this statement.

During the treatment, protons with energies in the range of 70 to 250 MeV are used. The higher the energy, the higher the maximum depth in tissue. To get the proton energies to this level, cyclotrons, synchrotrons and synchro-cyclotrons are used. (Mohan and Grosshans 2016)

2.2.2 Alternatives to protons

Beside protons, other ions are also investigated for their potential to replace photon therapy. Contenders with possibly suitable characteristics are deuterium (H^{2+}), Helium-3 (He^{1+}), alpha particles (He^{2+}) (Azizi and Mowlavi 2013; Bernal-Rodríguez and Liendo 2013; Bichsel 2013), and carbon ions (C^{6+}). (Bernal-Rodríguez and Liendo 2013; Bichsel 2013) The use of heavier ions offers the benefit of less scattering (Mohan and Grosshans 2016) and sharper Bragg peaks. (Degiovanni and Amaldi 2015) Some differences in Relative Biological Effectiveness (RBE) were also found. RBE is a measurement tool to compare the effectiveness of different forms of ionizing radiation, when the same amount of energy is absorbed. For Carbon-ions it was found that the RBE could be three times higher for some types of cells in comparison to protons. As a downside, heavier ions have to be accelerated to significantly higher energies to reach the same depth in tissue than protons. For example, a C^{6+} -ion needs an energy of 4800 MeV to travel through 28cm of water, while a proton only needs 210 MeV. This is more than 22 times as much. (Degiovanni and Amaldi 2015)

This does, however, not mean that it is not a feasible option. At the moment, there are multiple functional carbon ion facilities in the world.

2.2.3 Dose control

Most of the time, multiple sessions of therapy are needed to completely destroy the tumor. However, the dose of the dosage beam has to be corrected in subsequent sessions, depending on the dose deliveries of the earlier sessions. Even with the technology of proton therapy constantly evolving and the ability to create correct proton dose distributions more precisely to minimize damage to normal tissue, there is a problem. No body is the same, so even with extensive simulations and calculations, unanticipated events, like a large dose deposition in an organ, can take place. (McCleskey et al. 2015) Deviations due to the unforeseen effects occur, and succeeding dose deliveries have to be altered to take these anomalies into account. A patient can undergo a CT scan again, but this exposes the patient to extra radiation dose, damaging healthy tissue again. Luckily, techniques have been found to deal with this issue.

Different ways to solve this problem have been suggested. A distinction can be made between direct and indirect methods. Direct methods involve direct dose measurement. Examples are an implantable dosimeter which can be read off from a distance and proton radiography and tomography. (Knopf and Lomax 2013)

Indirect measurements are also possible, these methods use secondary particles that get released due to the irradiation of tissue. At the moment, the two main approaches are positron emission tomography(PET) and prompt gamma imaging(PGI). (Perali et al. 2014)

PET takes advantage of coincident gammas. These originate from pair annihilation, the opposite of pair production. When a proton travels through tissue, a possible interaction is the creation of ^{11}C , ^{13}N and ^{15}O . These isotopes emit positrons, and when these positrons interact with an electron of the surrounding tissue, two annihilation photons are emitted at the expense of the positron-electron pair.

On the other hand, PGI makes use of prompt gammas(PGs). These PGs originate from the tissue that gets irradiated. During the dose delivery, the nuclei in the tissue get excited subsequently PGs are emitted in the decay of the excited nuclei. A PGI detector tries to 'catch' these PGs and then creates an image of the origins of the PGs. One of these imaging techniques will be treated in the next chapter. In theory, this technique should lead to a reliable portrait of the route of the proton beam, but a reliable enough imaging system still does not exist (Knopf and Lomax 2013). All methods come with different advantages and disadvantages, but those will not be discussed in this report. The focus lies on the Compton Camera.

2.2.4 Energies of interest

During proton therapy, some PGs occur more than others. Most PG energies emitted by human tissue range from 2 – 15 MeV(Knopf and Lomax 2013). When checking the energy spectrum, some notable peaks in occurrence rate can be seen. In *figure 8*, which is based on a simulation in GEANT4 with proton energies ranging from 100.83 MeV to 121.20 MeV, 4 peaks stand out.

- 0.511 MeV, which is released due to positron-annihilation
- 2.223MeV(Hydrogen), which can be explained as the photon emitted during neutron capture of hydrogen(Downey and Sandy 1986; Krimmer et al. 2018)
- 4.438MeV(Carbon) originates from excited ^{12}C returning to the ground state(Azizi and Mowlavi 2013; Kelleter et al. 2017; Krimmer et al. 2018; Perali et al. 2014).
- 6.129MeV(Oxygen)originates from excited ^{16}O returning to the ground state(Azizi and Mowlavi 2013; Kelleter et al. 2017; Krimmer et al. 2018; Perali et al. 2014).

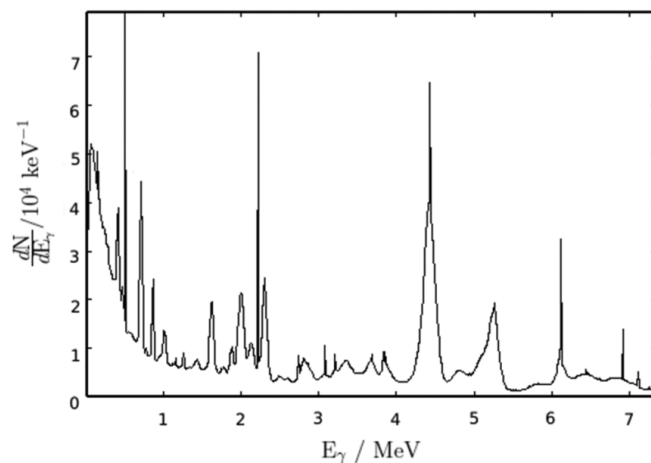


Figure 8: PG emission spectrum during proton therapy (source: H. Rohling)

Notable is that these last three elements are also the atoms that contribute most to the mass of human tissue. When looking at weight, the human body consists for 65% oxygen, 18.5% carbon and 9.5% hydrogen. This makes for a total of 92.5% of total human weight. If Nitrogen, the fourth biggest element, is added to the list, a total average of 96% of the weight of human tissue is reached(Saeed et al. 2016). The most common photons emitted by excited nitrogen are the ones with energies of 10.829 MeV. The underlying process is neutron capture by the particle (Chung, Wei, and Chen 1993).

2.3 Compton Imaging

2.3.1 Compton Camera

A Compton Camera(CC) is a detector specially made to detect Compton scattering, as its name already suggests. The fields of astronomy, solid-state physics(Knopf and Lomax 2013), homeland security, nuclear medicine(Jan, Lee, and Huang 2017) and factory safety(Park et al. 2014; Sinclair et al. 2014) use Compton cameras already for a range of reasons, some even for a few decades already. However, its employability in proton therapy has only been proposed in 2006(Knopf and Lomax 2013) and there is still a lot to be improved.

The main task of the Compton Camera is to acquire the data needed to find the location of a PG source. This is done by creating so-called Compton Cones. The surface of such a cone contains all the possible trajectories a photon could have taken before entering the Compton Camera. One cone does not narrow down the location of the gamma ray source since any place on the surface of the cone could be correct. Creating a Compton Cone for a second photon already shrinks down the possibilities. Assuming that the photons originate from the same source, the source should be located at the intersection of the two cones. Adding more cones, the location can be narrowed down with even better accuracy. The detectors do not create these cones by themselves, but they generate the data needed to do so.

Compton Cameras are able to detect the coordinates and energies of Compton scattering of photons. One interaction per photon is not enough to create a Compton Cone. In some cases, two interactions give enough data, but in most cases three interactions are needed. This can be seen in *figure 9*. First of all, the following formula can be used to calculate the initial energy in case three interactions are detected:

$$E_0 = \Delta E_1 + \frac{1}{2} \left(\Delta E_2 + \sqrt{\Delta E_2^2 + \frac{4\Delta E_2 m_e c^2}{1 - \cos \theta_2}} \right) \quad [1]$$

ΔE_1 and ΔE_2 are the energy deposition of the photon during the first interaction with location $p_1 = (x_1, y_1, z_1)$ and the second interaction $p_2 = (x_2, y_2, z_2)$. These two energies are directly measured by the Compton Camera. $\cos \theta_2$ can be determined using p_2 and p_3 . A visual representation can be seen in *figure 10*. Using the calculated value for E_0 , the energy of the incoming PG, the apex angle of the cone can be calculated using the following formula:

$$\cos \theta_1 = 1 + m_e c^2 \left(\frac{1}{E_0} - \frac{1}{E_0 - \Delta E_1} \right) \quad [2]$$

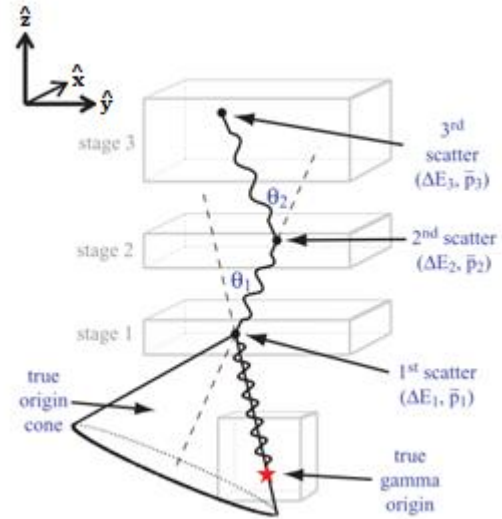


Figure 9: The necessary data for Compton imaging

With the knowledge that the apex of a cone can be found at the point of first interaction with the CC, the cone can be created. Repeating this process for all data, a number of cones is created equal to the amount of photons detected by the CC. Mapping all cones

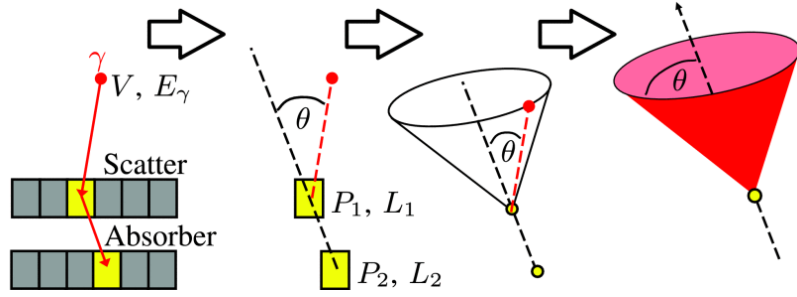


Figure 10: Creation of a Compton Cone (source: S. Schoene)

and checking the intersections can give an accurate prediction of the location of the PG source.

2.3.2a Two stage Compton Camera

As mentioned before, sometimes a PG only has to interact two times with the CC. This is the case if the photon is fully absorbed during the second interaction. With this method, the initial PG energy can be determined directly, since in this case $E_0 = \Delta E_1 + \Delta E_2$ (MacKin et al. 2013; McCleskey et al. 2015; Schoene et al. 2017). Two-Stage CCs try to fully absorb the photons during the second interaction.

Two-stage CCs consist of two detector crystals: a scatterer and an absorber. The functions are almost self-explanatory: The role of the scatterer is to catch a Compton scattering interaction and registrate its coordinates and energy deposition. The absorber also determines the coordinates and energy of the PG, but instead of Compton scattering, the necessary interaction is the photoelectric effect, 'absorbing' the photon. Formula [2] can then be used to form the Compton Cone. The absorber crystal often has a larger thickness than the scatter crystal, to increase the probability of photon absorption.

2.3.2b Multistage Compton Camera

Another widely used type of Compton Camera is the multistage CC. (MacKin et al. 2013; McCleskey et al. 2015; Schoene et al. 2017) Instead of a scatterer and an absorber, it has multiple scatterers, most often two, and an absorber. An advantage of this model is that the photon does not have to be absorbed by the absorber, it just has to have a detectable interaction with the crystal. Looking at Formula [1], this is easily seen. On the assumption that the PG Compton scattered in both scatterers, ΔE_1 and ΔE_2 are known. $\cos \theta_2$ is the last unknown value. However, using p_2 , p_3 and trigonometry, θ_2 can be found.

2.3.2c Single Stage Compton Camera

Another possibility is a Singlestage Compton Camera. It consists of only one crystal. The crystal functions as both scatterer and absorber. To get the necessary data, a PG has to undergo Compton scattering followed by photoelectric effect, or two Compton scatterings succeeded by another detectable interaction. These are the processes needed for twostage CCs and multistage CCs, respectively.

2.3.3 Cones

The Merriam-Webster dictionary defines a cone as ‘a solid bounded by a circular or other closed plane base and the surface formed by line segments joining every point of the boundary of the base to a common vertex’ (Cone | Definition of Cone by Merriam-Webster n.d.). This is a great way to describe the object, however, for scientific use, completely useless. A better way to describe a cone would be a mathematical definition.

A good start is a round base, defined by the following formula:

$$x^2 + y^2 = C$$

This creates a circle with a constant radius $r = \sqrt{x^2 + y^2}$. To create a non-circular base, parameters a and b are added:

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = C$$

Now let us add a third dimension, and place an axis through $x = y = 0$. This axis is perpendicular to the xy -plane. On this axis, choose a point z , this will be the z -coordinate of the cone. The apex can be found at the origin ($x = y = z = 0$). To create the cone, the borders of the round base have to be connected to the apex. Now that the cone is complete, it would be interesting to know the apex angle. Since z and r are perpendicular and their value is known, this can be done with a simple *tan*-relation:

$$\tan(u) = \frac{r}{z} = \frac{\sqrt{x^2 + y^2}}{z}$$

In this relation, u stands for the apex angle. x and y are coordinates located on the cone surface and z is the corresponding location on the cone axis of the angle. The formula can be rewritten in the following way:

$$x^2 + y^2 = \tan^2(u) \cdot z^2$$

Angle u is also known as the apex angle and has the range $u = \left(0, \frac{\pi}{2}\right)$. For $u = 0$, $\tan^2(u) = 0$ and the cone becomes a line. For $u = \frac{\pi}{2}$, $\tan^2(u) = \infty$ and the cone turns into a plane. The above formula is the basic formula for a cone with an apex at the origin. However, a more global notation is possible. If the apex of the circle is found somewhere else than at the origin (x_0, y_0, z_0) , the formula takes the following form:

$$(x - x_0)^2 + (y - y_0)^2 = \tan^2(u) \cdot (z - z_0)^2$$

This equation is valid for all ‘right circular’ cones. Right circular means that the basis of the cone is circular and the apex is located above the center of the basis.

Another expansion of the formula will be to look at cones with an elliptic base. The ratio between the maximum x - and y -values will not be $\frac{x}{y} = 1$, but it will be equal to some other constant $\frac{x}{y} = \frac{A}{B}$. If the ratio $\frac{x}{y} < 1$, the maximum in the y -direction is larger than maximum in the x -direction. In

this case, $y_{max} - y_0$ is called the major axis and $x_{max} - x_0$ the minor axis. If $\frac{x}{y} > 1$, $x_{max} - x_0$ is called the major axis and $y_{max} - y_0$ the minor axis. To take this into account, the cone formula transforms to this form:

$$\frac{(x - x_0)^2}{A^2} + \frac{(y - y_0)^2}{B^2} = \tan^2(u) \cdot (z - z_0)^2$$

Compton Cones are all right circular, so expanding the formula to larger proportions to take into account other variations of cones is not necessary.

2.2.4 Conic Sections

While researching cones, an interesting phenomenon to study is the intersection between a cone and a plane. If the crossing occurs at any other place than at the apex, the intersection will be a line, but the shape of this line depends strongly on the angle between the cone axis and the plane. Four different shapes are possible: A circle, an ellipse, a parabola and a hyperbola. This can be seen in *figure 11*.

2.2.4a Circle

In case of a circle, the angle α between the plane and the cone axis equals $\alpha = \frac{\pi}{2} = 90^\circ$. The size of the circle representing the conic section equals

$$(x - x_0)^2 + (y - y_0)^2 = \tan^2(\theta) \cdot (z - z_0)^2$$

In this equation, θ stands for the apex angle. In the case of a circle, the value of θ does not matter.

2.2.4b Ellipse

Figure 12 shows a conic section forming an ellipse. For the conic section to take the shape of an ellipse, α and θ both start to play a role. To form an ellipse, $0 < \alpha < \frac{\pi}{2}$ and $\theta + \alpha < \frac{\pi}{2}$. In words, this means that after the plane crosses one side of a cone, the plane also has an intersection with all other 'line segments' of the cone. With the above conditions, all 'line segments' of the cone cross the plane, creating the ellipse with the formula:

$$\frac{(x - x_0)^2}{A^2} + \frac{(y - y_0)^2}{B^2} = \tan^2(\theta) \cdot (z - z_0)^2$$

As discussed before, the values of a and b depend on the ratio between the major and minor axis. The formula can be simplified by bringing the right side to the left side. This leads to the following ellipse equation:

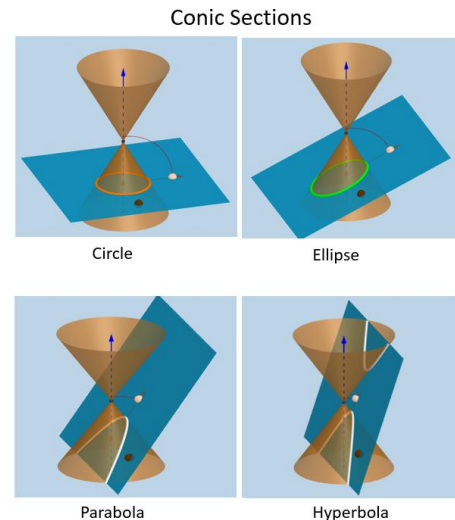


Figure 11: All four possible conic sections(source:<https://www.onlinemathlearning.com/conic-parabolas.html>)

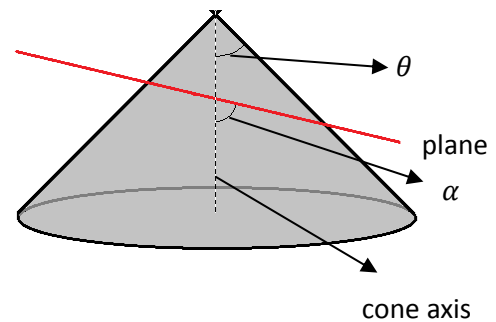


Figure 12: A conic section

$$\frac{(x - x_0)^2}{a^2} + \frac{(y - y_0)^2}{b^2} = 1$$

$$a^2 = A^2 \cdot \tan^2(\theta) (z - z_0)^2$$

$$b^2 = B^2 \cdot \tan^2(\theta) (z - z_0)^2$$

This formula is known as a standard ellipse formula.

2.2.4c Hyperbola

Another potential shape for a conic section is a hyperbola. In this case, $0 < \alpha < \frac{\pi}{2}$, just like with the ellipse, but the difference is that $\theta + \alpha > \frac{\pi}{2}$. This means that after the plane intersects with one side of a cone, the plane diverges with the other side of the cone, meaning they will never intersect. The resulting shape is a parabola. However, in the case of a double cone, like in figure 9, there will be another intersection, since the plane will also cross the opposite cone. This intersection will have the same shape. Together they form an hyperbola.

2.2.4d Parabola

Another potential shape for a conic section is a single parabola. This conic section is a special case and only occurs when the $\theta + \alpha = \frac{\pi}{2}$. After the plane intersects with a part of the cone, the plane lies parallel to the opposite part of the cone, meaning they will never intersect. Up to this point, it looks a lot like a hyperbola. However, the difference is that the plane also lies parallel to the second cone in the case of a double cone. The plane will never intersect with this cone since it also lies parallel to the closest side of the cone. This means they will also never intersect.

2.2.5 Detector Material

Many semiconductors are potentially suitable to be used as detector crystals in a Compton Camera. Scientists are still figuring out which material is most suitable for which situation, since every crystal comes with its own positive and negative sides. However, there are some guidelines for finding the correct detector materials. These guidelines also help to immediately know if a semiconductor is not suitable.

Parameters include the effective atomic number (the amount of protons an electron 'sees' for a material) of the semiconductor in the detector, the effective electron density, and the mass attenuation coefficient (the interaction probability per unit length) (Singh and Badiger 2016), but the detectors also need to be suitable for the energy range (In the case of medical application, 2-15 MeV (Knopf and Lomax 2013))

The more photons a detector is able to catch, the more data can be generated. The Lambert-Beer law is a useful tool to investigate the ability of a medium to do so. The law describes the rate at which the intensity of a beam of photons decreases when travelling through matter, also known as the attenuation of a beam. The relation is shown in the following formula:

$$I = I_0 e^{-\mu_m t}$$

In this relationship, I_0 is the incident photon intensity, I is the attenuated photon intensity, $\mu_m = \mu/\rho (cm^2 g^{-1})$ is the mass attenuation coefficient and $t = \rho x (g \cdot cm^{-2})$ is the mass per unit area. (Singh and Badiger 2016).

3. Materials and methods

3.1 Monte Carlo Simulations

For the research, data was gathered by another student. For the Monte Carlo simulations, Monte Carlo N-Particle eXtended(MCNPX) was used. This software can be used to simulate various processes involving photons, electrons, neutrons, or combinations of these three fundamental particles.(Los Alamos National Laboratory: MCNP Home Page n.d.)

The function of the code most relevant for our research was the one called PTRAC(Particle TRACK). The code was used to create a PG source and subsequently simulate the interactions in a detector. After translation of the created data file, the fate of every individual photon can be seen. As mentioned before, a photon can undergo various processes while interacting with matter: Compton scattering, the photoelectric effect, pair production and elastic scattering. The occurrence of these four events can be counted, while simultaneously determining the location of the event in three dimensions. The data can be used to study the effectiveness of the interactions of various photons in detectors.

In our case, PTRAC was used to test the efficiency of a few models of Compton Camera's over a range of different energies. This was done by simulating a single point source of gamma radiation and a Compton Camera with the desirable shape. The distance between the radiation emitter and the center of the crystal was set to 2 meters. The simulations were set up in such a way that the point source only emitted photons that directly hit the detector, forming a conical beam. Every simulation consisted of 10,000 photons.

The simulations were done with two different setups. During the first setup, the photons entered the detector on the circular surface. Afterwards, simulations were done with the beam aimed at the side of the detector. This can be seen in *figure 13*.

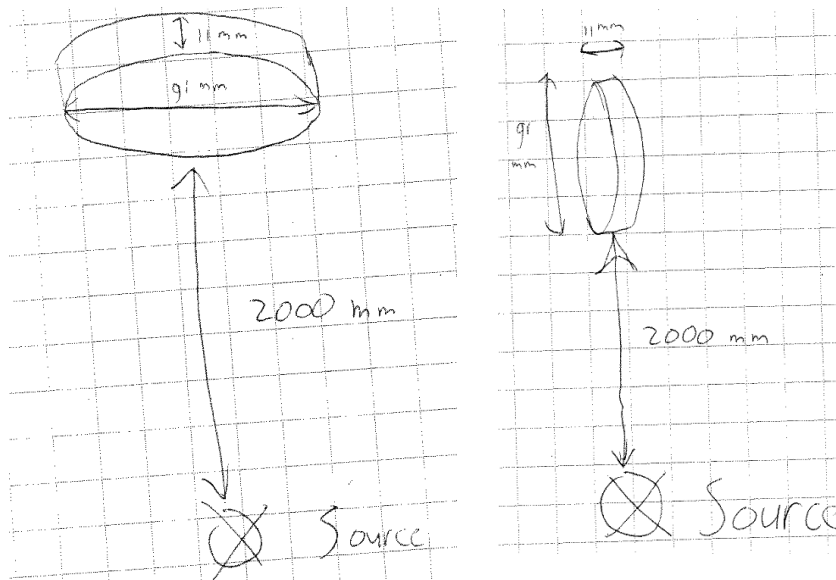


Figure 13: Setup 1(left) and Setup 2(right)

There was, however, a small difference between the data acquired for the two setups relevant for this paper. For the simulations done for the second setup, the coordinates of p_3 were recorded. For setup 1, they were not. More details and information can be found in (Pilarski 2017). The amount of useful data generated per simulation is shown in *table 1*.

3.2 Detector size and material

The dimensions of the detector studied are based on the GeGI detector manufactured by PHDS. (GeGI Info Sheet - PHDS Co. - Germanium Gamma Ray Imaging Detectors n.d.) The crystal in this detector is cylindrical, with a diameter of 91mm and a thickness of 11mm. The used material was Germanium(Ge). The detector has a spatial resolution of $\Delta x = 1.5 \text{ mm}$ and an energy resolution of $\Delta E = 2.1 \text{ keV}$.

	Energy (MeV)	# of caught PGs
Setup 1	2.223	191
	4.438	112
	6.129	82
	10.829	51
Setup 2	2.223	1014
	4.438	588
	6.129	422
	10.829	236

Table 1: The amount of suitable PGs for every simulation

3.3 Matlab

To investigate the efficiency of the two setups by analyzing the data simulated by Pilarski(Pilarski 2017), some tools had to be created. The choice was made to create these tools in Matlab, a programming language especially focused on numerical computing. It offers a lot of options that could come in handy during this research project. While Matlab offers the ability of manipulate matrices, has multiple plotting options for data, and has the possibility to test algorithms immediately, it also has an active community that shares a lot of handy programs for free and that tries to help you when you run into some trouble.

The biggest goal for the research was to see if it was possible to backtrack the path of the simulated photons and to create an image of the gamma source. First of all, since the location of the gamma source was known, it was not necessary to simulate all Compton cones. Instead, the conic sections with the xy-plane at $z = z_0$ were programmed. This resulted in one ellipse for every triple Compton scattered PG. A plot of the ellipses can be shown on a grid with variable size. In this paper, everything is done in a 'area of interest' with $x = [-25 \ 25]$ and $y = [-25 \ 25]$, divided into a window of pixels. The used length-scale was centimeter. The size of the pixels can also be changed. Subsequently, a program was created that filters out every ellipse that does not show up in the grid. The next program on the list was able to track the curves of the ellipses and calculate the distance it covered in every pixel. And finally, a code was written to create an image of the intersections of the ellipses, with the value of every pixel equal to the distance covered in that pixel, revealing the highest probable location of the PG source. While it was thought that the Matlab community already created some tools to make the path a little easier, none were found, so everything had to be coded by hand.

Another goal of the codes was that the codes are able to treat the data as one complete matrix D , in the form of $D = [x_1 \ y_1 \ z_1 \ x_2 \ y_2 \ z_2 \ x_3 \ y_3 \ z_3 \ \Delta E_1 \ \Delta E_2]$. $[x_1]$ stands for the vector consisting of all values of x_1 for that particular simulation, $[y_1]$ stands for the vector containing all values of y_1 , and so on.

3.4 Cones and ellipses

With the data from the Monte Carlo Simulations, consisting of $p_1 = (x_1 \ y_1 \ z_1)$, $p_2 = (x_2 \ y_2 \ z_2)$, $p_3 = (x_3 \ y_3 \ z_3)$, ΔE_1 , ΔE_2 , it was possible to take two different routes to find the Compton angle.

Since the data was acquired due to a simulation, the exact E_0 was known, so this value could just be plugged in into formula [2]. However, it would also be interesting to see what would happen if E_0

would be calculated with formula [1] and subsequently compare the results. This is only possible if θ_2 is known. This is the angle between the photon trajectory between the first and second interaction (D1->D2) and the photon trajectory between the second and third interaction(D2->D3). With some calculations and the coordinates of p1, p2 and p3, this angle can be determined with formula [3].

$$\hat{u} = \frac{u}{|u|} \rightarrow \hat{x}_1 = \frac{x_2 - x_1}{r_1} \text{ with } r_1 = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}$$

$$\cos \theta_2 = \frac{\hat{x}_1 \cdot \hat{x}_2 + \hat{y}_1 \cdot \hat{y}_2 + \hat{z}_1 \cdot \hat{z}_2}{1} \quad [3]$$

The next point on the list was to make a plot of the ellipses created by the conic section between the Compton Cone and the xy -plane at $z = -200\text{cm}$. A possible way to do this is to first find the location of the middle of the ellipse and to next build the ellipse around this point. To form the conic section, the middle of the ellipse and the lengths of the major and minor axes need to be determined. This was done by first determining φ_z , the angle between the photon trajectory and the z -axis at D1->D2, and φ_x , the angle of the photon trajectory with respect to the xy -plane at D1->D2. This can be seen in *figure 14*.

$$\cos \varphi_z = \frac{z_2 - z_1}{r_1} \quad [4]$$

$$\cos \varphi_x = \frac{x_2 - x_1}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}} \quad [5]$$

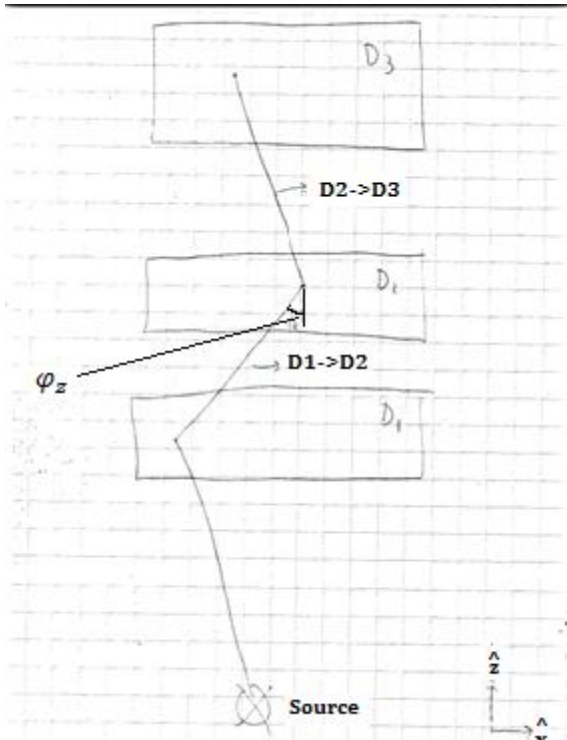
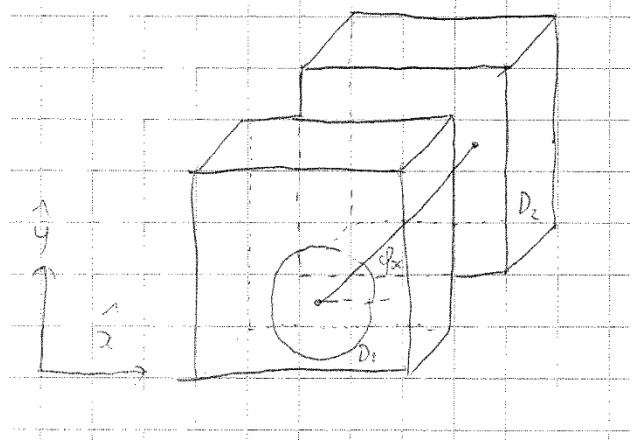


Figure 14: Determination of φ_z and φ_x



With these values, it is achievable to calculate the x - and y -coordinates of the middle of the ellipse, x_0 and y_0 .

$$x_0 = x_1 + |z_1 - z_0| * \tan(\varphi_z) * \cos(\varphi_x) \quad [6]$$

$$y_0 = y_1 + |z_1 - z_0| * \tan(\varphi_z) * \sin(\varphi_x) \quad [7]$$

Of course, an ellipse has a major axis and a minor axis. To find the lengths of the major axis, a , and the minor axis, b , the following formulas were used:

$$a = |z_1 - z_0| * (\tan(\varphi_z) - \tan|\varphi_z - \theta_1|) \quad [8]$$

$$b = |z_1 - z_0| * |\tan \theta_1| \quad [9]$$

Now that all necessary angles and lengths are known, the ellipses can be plotted as a function of t , the azimuthal angle. t is introduced to follow the ellipse line. This gives the following equations:

$$X = x_0 + a * \cos(\varphi_x) * \cos(t) - b * \sin(\varphi_x) * \sin(t) \quad [10]$$

$$Y = y_0 + a * \sin(\varphi_x) * \cos(t) + b * \cos(\varphi_x) * \sin(t) \quad [11]$$

t can be any value, but $0 = t = 2\pi$

When plotting functions in Matlab, the program calculates the values for (in this case) X and Y and puts these values into a graph. However, in contrast with a normal plot, a computer cannot make the plot continuous, but calculates a number of values for (in this case) X and Y and puts these in a graph. Matlab will draw lines between these points and this will be the shown image. Since every value for t gives a X - and a Y -value, the amount of steps taken between 0 and 2π is important. When the amount of steps is too small, the plot will not be very realistic since curves will not be smooth due to too many visible edges, but when the amount of t -values is too large, the calculation will be slower than necessary, while not significantly improving the graph. In this paper, t is a vector with a length of 100, ranging from 0 to 2π . The size of the steps is constant.

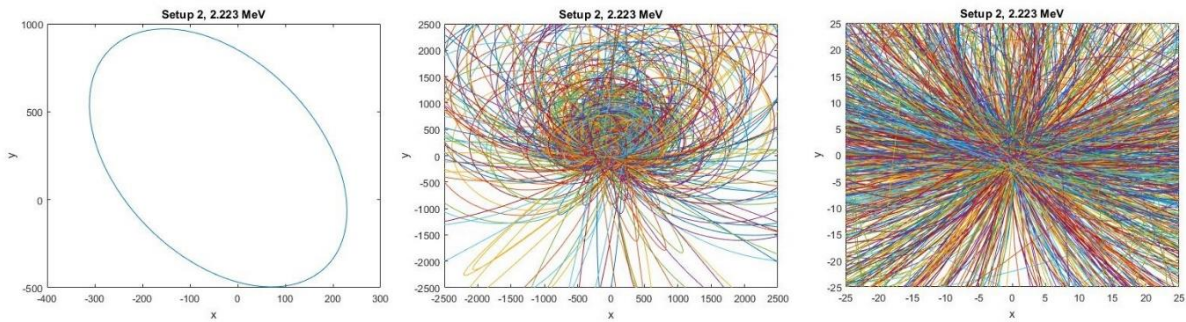


Figure 15: A plot of all elliptic conic sections for setup 2, with energy 2.223 MeV

One hundred steps should be enough to make a reliable image, as can be seen in figure 15. The image on the left shows the creation of one ellipse, the image in the middle shows all ellipses for a particular energy and setup, and the last image is the same as the one in the middle but the image size is 25 by 25 centimeters. It is clearly visible that the hotspot for the ellipses is around the point $[0,0]$. This is as expected, since the source was chosen to be at this location.

3.5 Grids

To track the ellipses while they cross the grid, it is important that a good starting point on the ellipse is used. The ideal starting point is a place where an ellipse enters the grid, since the only thing that has to be done after finding this point, is following the line in one direction until it leaves the grid.

In this section, a method is discussed to find these coordinates. The code could be used for another nifty feature. Since it finds all ellipses with intersections with the grid, it can also find the ellipses that do not cross the area of interest. Creating another data matrix without all the data that does not show up in the plot is a way to increase computing speed. A decrease of data leads to less calculations, which leads to faster image creation. In case of this simulation, the ellipses of almost all data points cross the square with $x = [-25 \ 25]$ and $y = [-25 \ 25]$. For simulations, a lot of detected photons not originating from the source of interest, this can cut out a lot of unnecessary data. While just plotting ellipses takes almost no time, the tracking of the ellipses (treated later) is more time-intensive and decreasing the amount of ellipses can significantly increase the computing speed.

In our case, a method was used that tries to find the intersection of an ellipse with the borders of the image, $x = x_{min} = -25$, $x = x_{max} = 25$, $y = y_{min} = -25$, or $y = y_{max} = 25$. For example, to determine if the ellipse crosses $x = x_{max} = 25$, the following equation has to be solved:

$$X = x_{max} \quad [12]$$

$$x_0 + a * \cos(\varphi_x) * \cos(t) - b * \sin(\varphi_x) * \sin(t) = x_{max} \quad [13]$$

Written in the form $f(x) = 0$, this becomes:

$$f_{x_{max}}(t) = x_0 + a * \cos(\varphi_x) * \cos(t) - b * \sin(\varphi_x) * \sin(t) - x_{max} = 0 \quad [14]$$

The other three equations are:

$$f_{x_{min}}(t) = x_0 + a * \cos(\varphi_x) * \cos(t) - b * \sin(\varphi_x) * \sin(t) - x_{min} = 0 \quad [15]$$

$$f_{y_{max}}(t) = y_0 + a * \sin(\varphi_x) * \cos(t) + b * \cos(\varphi_x) * \sin(t) - y_{max} = 0 \quad [16]$$

$$f_{y_{min}}(t) = y_0 + a * \sin(\varphi_x) * \cos(t) + b * \cos(\varphi_x) * \sin(t) - y_{min} = 0 \quad [17]$$

3.6 Newton-Raphson Method

Since Matlab does not have a built-in function to solve these kinds of equations, other root-finding techniques have to be used to find the solutions. This does not give the exact answer, but an approximation is found as a result of numerical analysis. In this case, the Newton-Raphson Method is used. The method is based on a Taylor series (Otto 2015). For the method, an estimated guess of the the location of the root is needed. Let us call this location $f(x)$. Let us say that the actual root can be found at $x + \epsilon$, so $f(x + \epsilon) = 0$. We find that the Taylor expansion for $f(x + \epsilon)$ looks as follows:

$$f(x + \epsilon) = f(x) + f'(x)\epsilon + f''(x)\epsilon^2 + \dots \quad [18]$$

Since the technique requires a guess close to the actual root, this means that ϵ is small. This makes second- and higher order terms almost insignificantly small, so the expansion can be reduced to:

$$f(x + \epsilon) \approx f(x) + f'(x)\epsilon \quad [19]$$

Since $f(x + \epsilon) = 0$, this can be rewritten as:

$$\epsilon = -\frac{f(x)}{f'(x)} \quad [20]$$

As a result, a n -iterative scheme can be formed:

$$x_{n+1} = x_n - \frac{f(x)}{f'(x)}, \quad n = 0, 1, 2, \dots \quad [21]$$

$$\text{with } f'(x) = \lim_{\delta \rightarrow 0} \frac{f(x + \delta) - f(x)}{\delta} \quad [22]$$

The value of δ that is generally suitable is $\delta = 10^{-6}$ (Otto 2015), for our problem this also is usable value. With enough iterations, this method should give an accurate enough value for the root. In this case, five iterations are more than enough to find an value of t with an error smaller than 10^{-2} .

Keeping this in mind and plugging formula [22] into formula [21] gives the following equation:

$$x_{n+1} = x_n - \frac{10^{-6} * f(x)}{f(x + 10^{-6}) - f(x)}, \quad n = 0, 1, 2, 3, 4 \quad [23]$$

Plugging in the derivatives and formula [14] - [17] in formula [23] gives the following equations:

$$x_{max_{n+1}} = x_{max_n} - \frac{10^{-6} * f_{x_{max}}(t)}{f_{x_{max}}(t + 10^{-6}) - f(t)}, \quad n = 0, 1, 2, 3, 4 \quad [24]$$

$$x_{min_{n+1}} = x_{min_n} - \frac{10^{-6} * f_{x_{min}}(t)}{f_{x_{min}}(t + 10^{-6}) - f(t)}, \quad n = 0, 1, 2, 3, 4 \quad [25]$$

$$y_{max_{n+1}} = y_{max_n} - \frac{10^{-6} * f_{y_{max}}(t)}{f_{y_{max}}(t + 10^{-6}) - f(t)}, \quad n = 0, 1, 2, 3, 4 \quad [26]$$

$$y_{min_{n+1}} = y_{min_n} - \frac{10^{-6} * f_{y_{min}}(t)}{f_{y_{min}}(t + 10^{-6}) - f(t)}, \quad n = 0, 1, 2, 3, 4 \quad [27]$$

3.7 Intersections

The amount of intersections between an ellipse and one side of the square can be zero, one, or two. The total number of intersections with all four sides of the square can be zero, two, four, six or eight. For four, six and eight intersection points, $a \approx \frac{1}{2} * (x_{max} - x_{min})$ and $b \approx \frac{1}{2} * (y_{max} - y_{min})$. See *figure 14*. For six and eight crossings of the grid borders the coordinates $[x_0, y_0]$ have to be close to the middle of the roster in comparison to the grid size. Since the ellipses' major axes and minor axes are almost always significantly larger than the size of the area of interest, zero or two intersections are most common. This can also be seen in *figure 16*.

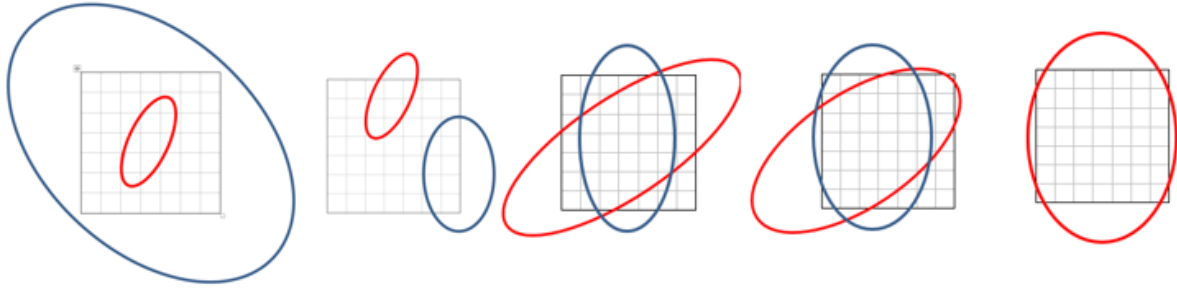


Figure 16: All possibilities for conic sections. From left to right: zero intersections, two intersections, four intersections, six intersections and eight intersections

For the program to work, an estimation of the value for t for the intersection was needed. If the value is too far from the actual intersection, the program gives an wrong value for t . Since it would be very time-intensive to estimate the t -value for every intersection, another method was used. Three initial guesses were taken, all three evenly distributed between the range of t ($= [1, 1 + 2\pi)$). The values chosen were

$$t = \left(1 + \frac{2\pi n}{g}\right), \text{ with } n = 1, 2, 3, \quad g = 3 \quad [28]$$

The 1 could be replaced by any other number. After determining values for t , the program calculates the matching X- and Y-coordinates immediately, putting them in a matrix U . For one point of data/ellipse/photon, this looked like table 2. Column 1 is the x-value and column 2 is the y-value. The program searches the intersection with $x_{max}/x_{min}/y_{max}/y_{min}$ around 3 different t -values, so 12 rows per ellipse makes sense. So Matrix U was of the size $[12 * \text{number of ellipses}, 2]$.

The next step was the selection process. First of all, the process looked through rows of U and if the following requirements were met, the two values in the row were placed in the first two columns of matrix T .

$$x_{min} \leq x_{n,1} \leq x_{max} \quad [29]$$

$$y_{min} \leq x_{n,2} \leq y_{max} \quad [30]$$

Matrix T had the dimensions $[12 * \text{number of ellipses}, 3]$. If none of the intersection points could be found in the grid, the first two columns of T were empty. The last column contained the t -value belonging to the x- and y-values that were now put in the columns. The part of the matrix representing the first ellipse can be seen in table 3. Keeping the t -value comes in handy during the tracking process, so keeping this in a matrix with the X- and Y-values of an intersection point gives a starting point for the tracking and an initial guess point for t . More

	1	2
1	25	-0.7375
2	-25.0000	3.6924
3	-68.7735	25.0000
4	-31.1945	5.5901
5	25.0000	270.6538
6	-24.9998	3.6923
7	92.4267	25.0000
8	-35.1404	6.9697
9	25.0000	270.6538
10	-25.0000	266.2244
11	-68.7735	25.0000
12	-59.6544	18.8652

Table 2: All twelve found intersection coordinates with the grid for one ellipse. Column 1 stands for the x-coordinate and column 2 the y-coordinate

Table 3: The twelve intersections found for one ellipse after filtering out all coordinates that do not lie into the grid

1	25	-0.7375	3.3228
2	-25.0000	3.6924	2.9526
3	0	0	2.5936
4	0	0	2.9050
5	0	0	6.2709
6	0	0	9.2357
7	0	0	3.8586
8	0	0	2.8744
9	0	0	6.2709
10	0	0	6.6412
11	0	0	8.8767
12	0	0	2.6742

about this can be found up ahead in the Tracking section.

Secondly, a matrix E was created to be an 1-to-1 copy of the original data matrix $D = [x_1 \ y_1 \ z_1 \ x_2 \ y_2 \ z_2 \ x_3 \ y_3 \ z_3 \ \Delta E_1 \ \Delta E_2]$. Subsequently, if the following statement was true, the corresponding row of E was deleted.

$$\sum_{k=1}^{12} \sum_{l=1}^2 |T_{(n+k),l}| \leq 0, \text{ for } n = 12, 24, 36, \dots, p - 12 \quad [31]$$

with p = total number of PGs/conic sections/ellipses

In words, if the first two columns of T were empty for all twelve rows, meaning that for that particular ellipse, no crossings with the borders of the grids were found on the basis of three different initial guess points, the representing row of data is deleted in E . The treated section of T is also deleted when it contains only zeros in the first two columns. This makes sure that all data in E is useful for the image creation of the area of interest.

In the last few sections, matrix U , T and E were created, each via their own functions. It would be easier to combine these matrices to one matrix with all relevant information again. Since E contains all data for ellipses that travel through the area of interest, let us use this matrix as a basis. The other relevant numbers are the X -, Y - and t -values, these can all be found in matrix T . Since all useless data is also removed from matrix T , the # of rows in $T = 12 * \# \text{ of rows in } E$. Since for every 12 rows, T included at least one set of coordinates of a point on the border of the area of interest and the accompanying t . The only thing that was left to do was extract these values and place them with their matching dataset in matrix E .

This was done by treating every set of 12 rows with data in matrix T separately. First of all, the rows in the set were sorted in ascending order based on the value in the first column. This is visible in *table 4*. If the conic section crossed x_{min} or entered the grid through y_{min} or y_{max} while x was negative, this coordinate would show up on in the first row, or at least above the rows that were set to zero. If this was not the case, the intersection happened when X was positive and the data would show up in the last row(s) of the set. Subsequently, after sorting, the code checked the following statement:

$$\sum_{n=1}^2 |T_{1,n}| \leq 0 \quad [32]$$

	1	2	3
1	-25.0000	21.2023	47.1165
2	-25.0000	21.2023	3.1342
3	-1.7220	25.0000	3.1425
4	-1.7220	25.0000	3.1425
5	0	0	-28.2639
6	0	0	0.8459
7	0	0	2.9756
8	0	0	3.1521
9	0	0	5.9448
10	0	0	5.9627
11	0	0	9.0919
12	0	0	15.1973

Table 4: All relevant intersections found in the grid, sorted in ascending order. Column 1 is the x-coordinate and column 2 the y-coordinate. Their t-value is also added in column 3.

If the statement was true, it meant that the first row of T contained coordinates of an intersection point. If the statement was not true, this meant there were no points of intersection with negative X -values, meaning the intersection happened in the positive- X region. Coordinates for a location at which the ellipse entered the grid could then be found in the bottom row. So if the statement was true, the first row of T would be added to form three new columns in matrix E , if the statement was false, the bottom row would be added to E . Since the first 12 rows of T all represented the first row in E , the second twelve rows of T represented the second row of E , and so on, the first extracted row of T could be added to the first row of E , the second row of T to the second extracted row of E ,

and so on until the end of the matrix was reached. Now E has all data that is needed for any relevant calculation, ellipse plots, tracking of the curve and image creation. An example can be seen in *table 5*.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	0.3222	-0.8906	0.5492	0.4263	-2.1197	-1.2653	0.3760	-2.4254	-0.3509	0.9546	1.0143	-25.0000	3.6924	2.9526
2	-0.1184	0.8358	3.6906	-0.2011	3.2243	1.9657	-0.3690	3.5918	1.7404	1.4303	0.0686	-25.0000	1.0874	3.0534
3	-0.2598	-1.5515	-2.5662	-0.1069	-1.2074	-2.7928	-0.1021	-1.2252	-2.8001	1.5074	0.4771	-25.0000	10.7161	3.0574
4	-0.0877	-1.7369	2.5692	-0.3605	-2.4993	0.9563	-0.4111	-2.6331	1.0143	0.7029	1.1132	-25.0000	-7.2419	2.8901
5	0.5184	-3.8267	1.6227	0.1954	-1.8866	1.7622	0.1718	-1.7970	1.8783	1.8301	0.0786	-25.0000	21.2023	47.1165
6	-0.1367	0.3180	2.0008	-0.1254	0.2881	2.2455	-0.0521	0.2770	2.2997	1.9930	0.0324	-25.0000	6.3781	8.2266
7	0.4143	-2.6105	-1.8694	0.3068	-2.7180	-2.1328	-0.4431	-3.0208	-3.0856	0.8182	0.1456	-25.0000	-21.7370	2.8624
8	0.0784	3.2130	3.0825	0.1215	2.5662	2.4721	0.4291	2.5142	2.4052	1.2843	0.5252	-25.0000	6.3697	3.0243
9	0.2617	1.8943	-0.4166	-0.5005	1.6713	-1.2767	0.0250	3.1641	-0.6635	1.1910	0.7921	-9.9484	-25.0000	2.9853
10	0.4970	3.6794	1.1726	0.3866	3.2874	-0.5186	-0.3643	2.7333	-1.5439	0.2398	0.7496	-25.0000	3.4703	2.6090
11	0.3998	-0.5850	3.3324	-0.0476	-0.5337	2.9651	0.1674	0.0218	2.7681	1.3683	0.5323	-3.6513	-25.0000	3.0423
12	-0.3897	3.2359	2.6200	0.1203	1.7298	2.4078	0.3684	1.4865	2.6433	1.7574	0.1107	-25.0000	11.8300	3.1244

Table 5: A part of matrix E.

The amount of t -values was set to three, but this could also be set to an integer of choice. However, when increasing the amount of guesses from 3 to 4 or higher, the program could not find any more ellipses that would cross the grid, so increasing this number would only increase computing time. Setting it to two would decrease the amount of correctly found intersections. Sometimes, it was only able to find one of the two intersections, however, this does not matter. For tracking, it is only necessary to find one root for every ellipse. This will be treated in the next section. The rest of the intersections can be found by tracking the function until another border of the grid is reached.

3.8 Tracking

In this part of the paper, a method to determine the highest probable location of the source of the PGs is discussed. To do so, a technique called ‘conic section tracking’ is used. The code I wrote is based on the method described by X. Lojacono in one of his papers (Lojacono 2014).

The code follows a conic section as it traverses the grid. The grid is split up in pixels, creating a roster. For the initial image creation, the pixel size was set to 2mm by 2mm, so 0.2 by 0.2. Once the Full Width at Half Maximum(FWHM) of the image of the source point is known, the size of the pixels can be increased or decreased if the FWHM is much larger or smaller than the pixel size. It is unnecessary to take a pixel size smaller than a twentieth part of the FWHM.

Since the grid lies in the range $x = [-25 \ 25]$ and $y = [-25 \ 25]$, the total size of the grid is $50 \text{ cm} \times 50 \text{ cm}$. The amount of pixels is 250×250 .

The conic section tracking code follows the curve of the ellipse as it enters the grid, travels from pixel to pixel, and finally leaves the grid. This can be seen in *figure 17*. First of all, a brief summary is given. The code first takes the starting coordinates out of matrix E. This starting point is on the edge of pixel in the edge of the grid. Subsequently, the program finds out which pixel border the curve crosses and what the corresponding X- and Y-values are. and the distance travelled through the pixel is calculated. Using the new coordinates, this process is repeated until another edge of the area of interest is reached. Then the method moves on to the next ellipse, until all ellipses are treated. All values are put in a $m \times 5$ -matrix $I = [x \ y \ d \ t \ \Delta t]$ with

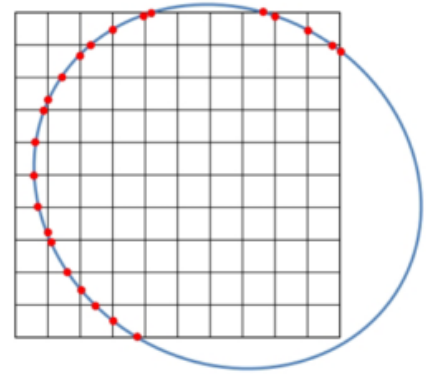


Figure 17: A visualization of the tracking script. (source: X. Lojacono)

$m =$ the total number of pixel edges crossed for all ellipses

$x =$ a vector with length m containing every x – coordinate of an intersection

$y =$ a vector with length m containing every y – coordinate of an intersection

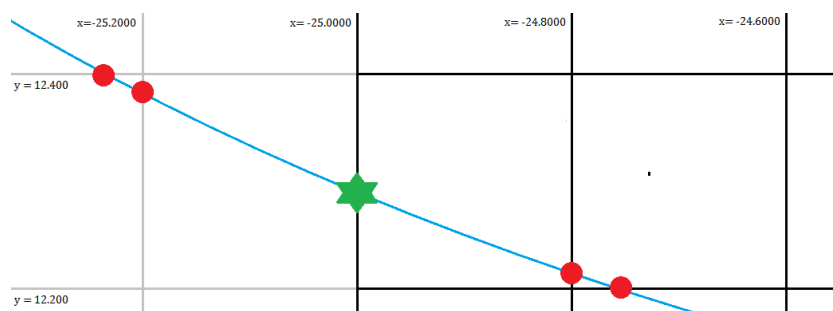
$d = \sqrt{(x_n - x_{n-1})^2 + (y_n - y_{n-1})^2}$, approximate distance travelled through pixel n

d is useful for the first step of the tracking code, but it also plays a big role in the image creation. Since the conic section is large for most PGs, the curvature is barely visible when looking at it in a $0.2 \times 0.2 \text{ pixel}$, so treating it as a straight line comes close to the real line. The importance of d will be discussed in the part **Image Creation**. Matrix I is also used again to create an image of the PG source in the next section.

Using the last three columns of E , containing the

coordinates and corresponding t-value of an intersection with the edge of grid, as a starting point, the first step to successful tracking was set. Since the program could not know the

Figure 18: Visualisation of the first step of the tracking script. The green star is the starting location



direction the curve would take into the grid, the following step involved investigating the surrounding pixel edges. No distinction could be made between the border at which the curve entered the area of interest, so all four directions were checked. For example, if the ellipse entered the grid at the location $[x\ y]=[-25\ 12.28]$. With a pixel size of 0.2×0.2 , Calculations were done to find the intersections with $x = -25.2000$, $x = -24.8000$, $y = 12.2000$ and $y = 12.4000$. The calculations were done using the Newton-Raphson method again, with equations looking like formula [24]-[27] but with the above values instead of x_{max} , x_{min} , y_{max} and y_{min} . A visual representation of the four calculated points can be seen in *figure 18*. This led to a X , Y , d , and $\Delta t =$

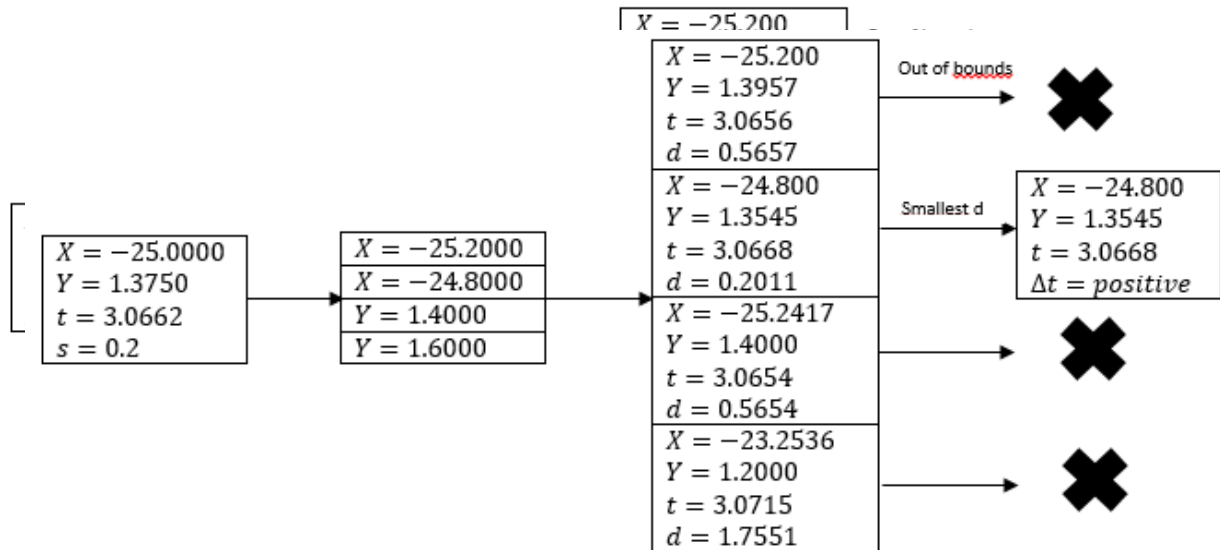


Figure 19: The selection process for tracking

$t_2 - t_1$ for all four intersections.

The program then took the crossing with the lowest value for d , while not taking account points that were located outside of the grid. This led to the location of the second point. With the data, the change in t was also calculated. This was a handy tool for error free determination of all other points, since for the tracking of one ellipse, t would always increase, or it would always decrease. Taking the value with the lowest increase/decrease in t would always give the intersection closest to the last one. The selection process can be seen in *figure 19*.

The next step in the process was easier, since it was known from which side the new pixel was entered, the code only had to check three intersections instead of four. All three were done at the same time. Subsequently, d was also determined for all three possible crossings. However, determining the shortest distance did not do the trick in this case. The reason for this is that sometimes, the distance to the crossing before the current intersection point is smaller than the real next border. This becomes clearer when looking at *figure 20*. It is a part of *figure 16*, but imagine that this time, the ellipse is tracked in the other direction. The green star indicates the current location. The code calculates the locations of intersection with $y = 12.2000$ (down), $y=12.4000$ (up) and $x = -25.0000$ (left). For the determination of the intersection value

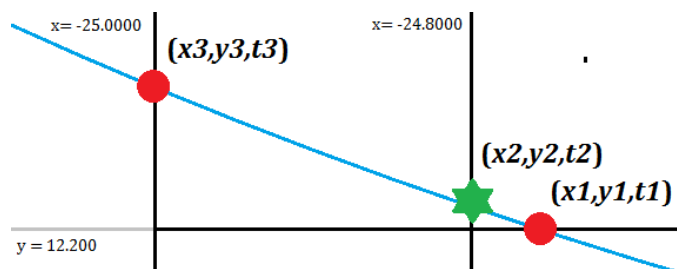


Figure 20: A visualisation of a possible problem that could occur

at the opposite side of the pixel, 0.2 was added or subtracted from the current value. The automatic determination of borders with pixel lengths of 0.2 was done by using the built-in Matlab tools $ceil(x)$, which rounds x up to the nearest integer, and $floor(x)$, which rounds down x to the nearest integer. To round up/down to the correct 1/5th integer, the following formulas were used:

$$X_n = ceil\left(\frac{x_{n-1}}{s}\right) * s \quad [33]$$

$$X_n = floor\left(\frac{x_{n-1}}{s}\right) * s \quad [34]$$

$$Y_n = ceil\left(\frac{y_{n-1}}{s}\right) * s \quad [35]$$

$$Y_n = floor\left(\frac{y_{n-1}}{s}\right) * s \quad [36]$$

However, there was another problem. While the correct new point should be the one to the left, the distance to the point with $y=12.200$ is shorter. If nothing else was done, the code would get stuck in an infinite loop between these two points, since the next step for the program would be to pick the location of the green star again.

To make sure this infinite loop would not occur, t comes into the picture. Since t will only increase or decrease while following a particular curve in the same direction, the following requirement had to be met to be eligible for the next coordinate in the tracking program:

$$\Delta t = t_n - t_{n-1} > 0, \quad \text{if } t_2 - t_1 > 0 \quad [37]$$

$$\Delta t = t_n - t_{n-1} < 0, \quad \text{if } t_2 - t_1 < 0 \quad [38]$$

$n = \text{the number of intersection points for one conic section in the area of interest}$

Out of the remaining possibilities, the option with the smallest Δt was chosen. Assuming no bugs in the calculations showed up, this should always lead to the correct intersection.

The process was repeated until the curve left the area of interest again. For every ellipse, the code only checks one possible curve that crosses the area of interest. However, there could be multiple curves belonging to the same ellipse that cross the edges of the grid, but these are not taken into consideration by the script. This is something that could be improved. More information about this problem can be found in **Code improvements**.

Subsequently, the process was repeated for all other ellipses that crossed the grid. As mentioned above, all relevant data was placed in a matrix $I = [x \ y \ d \ t \ \Delta t]$. Now all data was collected that was needed to create an image of the PG source. With help of these images we were finally able to compare the two different detector setups.

3.9 Image creation

The last step to create the image of the PG source using the data of the Compton Camera was to convert the values in matrix $I = [x \ y \ d \ t \ \Delta t]$ to points in the image. Before treating the method used for this process, the importance of d is first explained.

3.9a The distance d travelled through a pixel

When converting multiple continuous shapes to images, one has to choose how to handle the color of pixels. The color changes depending on the size of the value allocated to the pixel. Two options were considered:

- Adding the same value to all crossed pixels. For example, to stay in theme, picture two intersecting ellipses in a grid. All pixels start with a value of 0. While plotting the first ellipse, all crossed pixels get the value +1. When plotting the other ellipse in the same grid, +1 is added to the value of all pixels that contain this ellipse. Now all pixels without a ellipse on them have value 0, pixels which contain a part of a ellipse have value +1, and the pixels holding the intersections have value $(+1) + (+1) = +2$.

- Giving pixels value proportional to the distance the function travels through the pixel. Let us use the same example as above. Instead of giving a pixel a +1 every time an ellipse crosses it, the value of pixels containing a larger part of one of the ellipses increases more than the value of pixels with only a small part of one of the ellipses in them. If multiple conic sections cross a pixel, the value of the pixel equals the sum of all values allocated to it.

In our case, the second option seems like the best pick. The image created should give an impression of the location of the gamma ray source. As discussed before, the location of the PG can be found at an intersection of the surfaces of Compton Cones. Translating this to the conic section, taking the correct z (the z -coordinate of the gamma source, $z = -200$), which is luckily known in our case, the PG can be found on the intersection of the ellipses. The conclusion can be drawn that the probability of a pixel containing the PG increases if a larger part of a curve traverses it, since the chance that the pixel contains an intersection also increases. Giving pixels a value according to the sum of all d should be the best option to create an more precise image.

To match every d with its correct location, the only thing left to do is to convert X and Y to the coordinates of the matching pixel.

3.9b Matching d with the correct pixel

Since the range of the area of interest was $[x_{min} \ x_{max}] = [-25 \ 25]$ and $[y_{min} \ y_{max}] = [-25 \ 25]$ and the pixel coordinates start at $[1 \ 1]$, X and Y had to be converted suitable coordinates in the image plot. Since pixels of size $s \times s = 0.2 \times 0.2$ were initially used, the image pixel dimensions would become

$$image \ dimensions = \left[\frac{x_{max}-x_{min}}{s}, \frac{y_{max}-y_{min}}{s} \right] = \left[\frac{50}{0.2}, \frac{50}{0.2} \right] = [250, 250] \quad [39]$$

Converting the coordinates x and y to pixels also meant that a range of values were appointed to the same pixel. For example:

$$\begin{aligned} [x] = [-25] &\rightarrow [l] = [1], \text{ but also } [x] = [-24.81] \rightarrow [l] = [1] \\ [y] = [24.81] &\rightarrow [n] = [250], \text{ but also } [y] = [25] \rightarrow [n] = [250] \\ l &= \text{column number of pixel} \\ n &= \text{row number of pixel} \\ [x, y] = [-25, 25] &\rightarrow [n, l] = [250, 1] \end{aligned}$$

Notice that the X -value corresponds to the second value, l , in the pixel coordinate system. This can be explained by noticing that Matlab reserves the first input, represented as a n here, for the row number, and the second input, l , as the column number. This is the opposite of the generally used representation of an two-dimensional graph. In the latter, the value for the horizontal axis is listed first, and the value for the vertical axis is listed second.

The next action to be taken was to convert all x - and y -values matrix $I = [x \ y \ d \ t \ \Delta t]$ to the correct pixel coordinates $[n, l]$. First of all, let us introduce a function in Matlab to round up inputs called $\text{ceil}(x)$. For example, $\text{ceil}(1.9) = 2$. Using $\text{ceil}(x)$ again, the following formula was made:

$$(n_k, l_k) = \begin{cases} \begin{pmatrix} \text{ceil}\left(\frac{I_{k,2}}{S} + 10^{-8}\right) + 125 \\ \text{ceil}\left(\frac{I_{k,1}}{S} + 10^{-8}\right) + 125 \end{pmatrix}, & \begin{matrix} \Delta x \leq 0 \\ \Delta y \leq 0 \end{matrix} \\ \begin{pmatrix} \text{ceil}\left(\frac{I_{k,2}}{S} - 10^{-8}\right) + 125 \\ \text{ceil}\left(\frac{I_{k,1}}{S} + 10^{-8}\right) + 125 \end{pmatrix}, & \begin{matrix} \Delta x \geq 0 \\ \Delta y \leq 0 \end{matrix} \\ \begin{pmatrix} \text{ceil}\left(\frac{I_{k,2}}{S} + 10^{-8}\right) + 125 \\ \text{ceil}\left(\frac{I_{k,1}}{S} - 10^{-8}\right) + 125 \end{pmatrix}, & \begin{matrix} \Delta x \leq 0 \\ \Delta y \geq 0 \end{matrix} \\ \begin{pmatrix} \text{ceil}\left(\frac{I_{k,2}}{S} - 10^{-8}\right) + 125 \\ \text{ceil}\left(\frac{I_{k,1}}{S} - 10^{-8}\right) + 125 \end{pmatrix}, & \begin{matrix} \Delta x \geq 0 \\ \Delta y \geq 0 \end{matrix} \end{cases} \quad [44]$$

Since Δx and Δy are never 0 at the same time, it is safe to use the \leq or \geq for every requirement. The value 10^{-8} is taken because it is small enough that it will not influence the results. Taking the example based on *figure 15* again, with $\Delta x \leq 0$ and $\Delta y \geq 0$, the result is as follows:

$$l = \text{ceil}\left(\frac{x_3}{S} + 10^{-8}\right) + 125 = \text{ceil}\left(\frac{-25}{0.2} + 10^{-8}\right) + 125 = -124 + 125 = 1$$

$l = 1$, as it should be. All three other options can be checked easily using the same logic.

Using formula [44], all distances d can be linked to the correct pixel (n, l) for every pixel crossing k , creating a 250×250 – matrix $Z_{n,l}$. Every element of Z represented an pixel in the image, and the value the element contained was proportional to the sum of the length traversed through the pixel by all elliptic conic sections. Adding an x-axis with range $[x_{min} \ x_{max}]$ and y-axis with range $[y_{min} \ y_{max}]$ again completed the image. An example can be seen in *figure 21*.

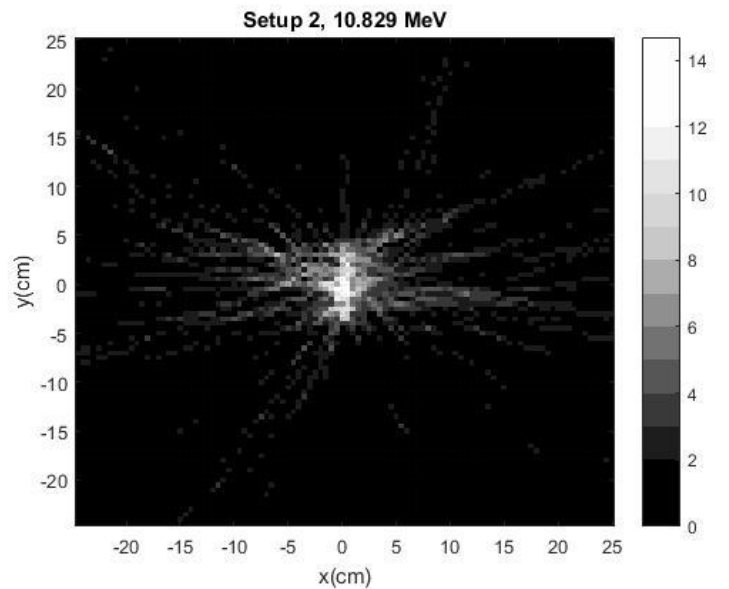


Figure 21: An example of a pixelated image. In this case, setup 2 was used and the PGs had an energy of 10.829 MeV

3.10 Full Width at Half Maximum(FWHM)

A small extra step was needed to determine the FWHM of the point source image. This would allow us to see if the pixel size $s \times s = 0.2 \times 0.2$ could be increased or decreased. This was done by selecting the row with containing the pixel with the highest value and the two rows next to it ($n = 124, 125, 126$) and summing the values of every column. Plotting these values in the range $x = [x_{min} \ x_{max}]$ gave a profile of the figure. The same was done using the column with the highest value-pixel and the columns adjacent to it and plotting these values in the range $y = [y_{min} \ y_{max}]$. This gave plots looking like *figure 20*. It also shows how the FWHM is determined.

For the simulation with the largest amount of collected data, The FWHM was investigated, since this should give the lowest influence of statistical fluctuations. The maximum was close to $d = 25$. The FWHM was more than 5cm. This lead to the adjustment to do all calculations using $s = 0.5 \times 0.5$. This decreased the calculation time immensely.

An calculation time comparison is shown in *table 6*. A comparison of the resulting plots is shown in *figure 22*.

	Energy (MeV)	Resolution s (cm)	Calculation time (s)
Setup 2	10.829	0.2×0.2	37.224
		0.5×0.5	11.465

Table 6: An comparison of calculation time for different values of s

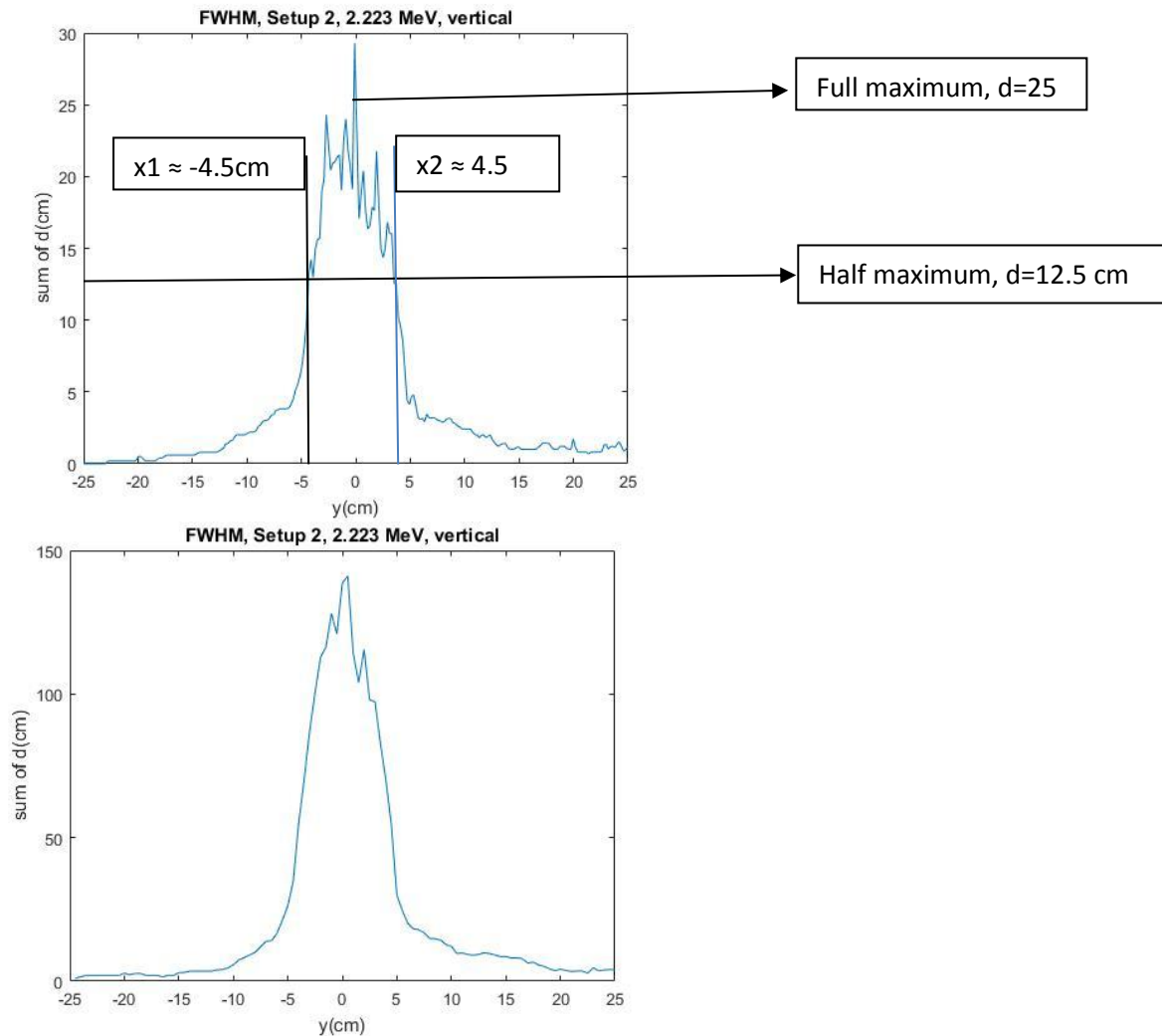
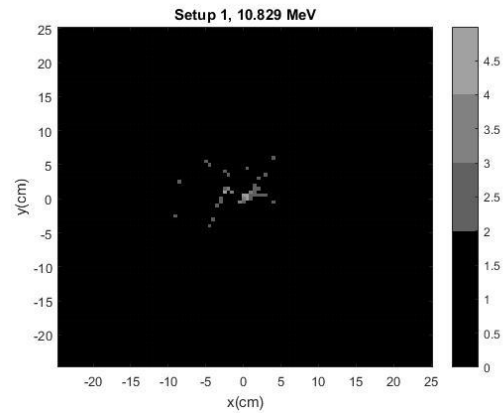
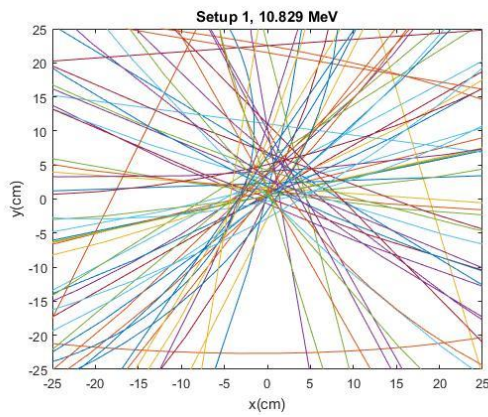
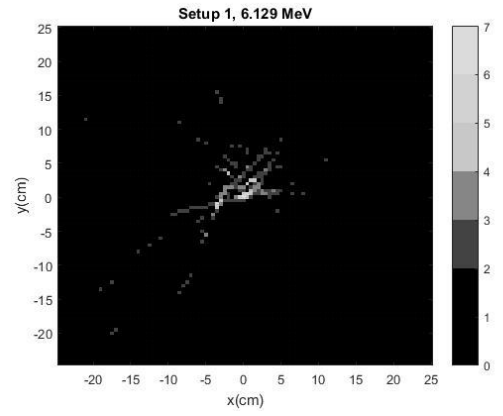
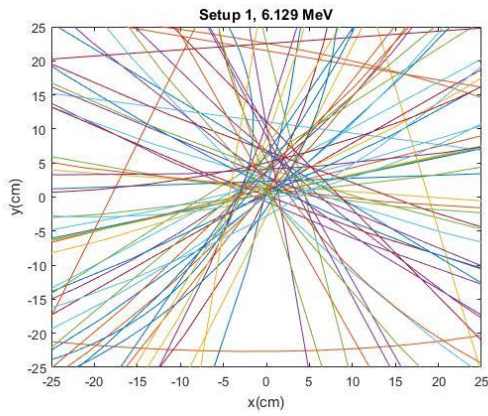
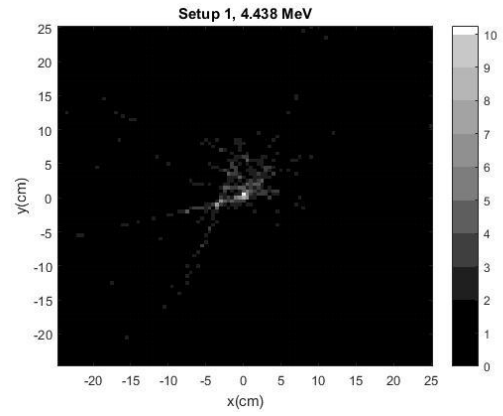
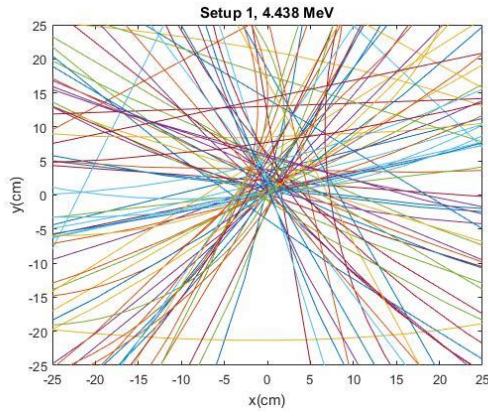
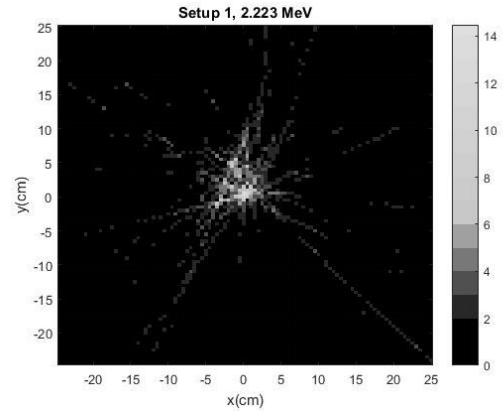
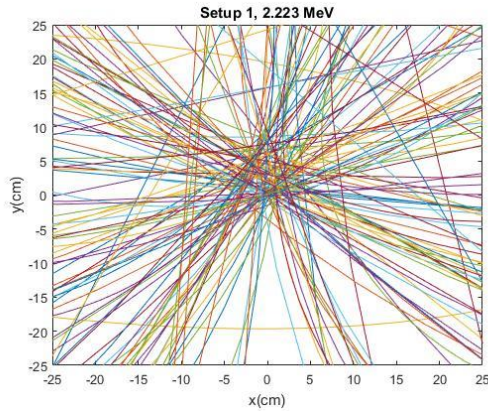


Figure 22: The plot used to determine the Full Width at Half Maximum. up is $s = 0.2 \times 0.2$ and down is $s = 0.5 \times 0.5$.

4. Results

First of all, in *figure 23 and 24*, all ellipses and their corresponding image are shown.



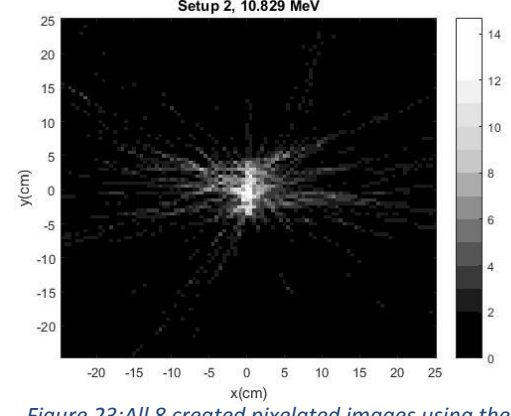
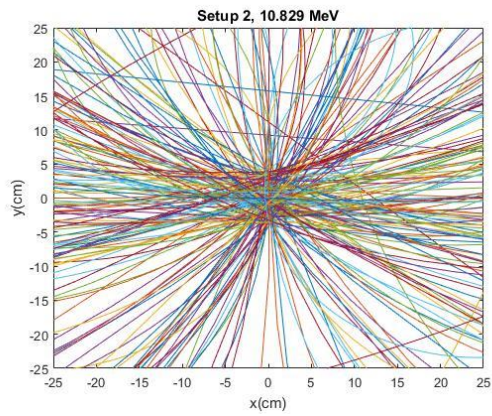
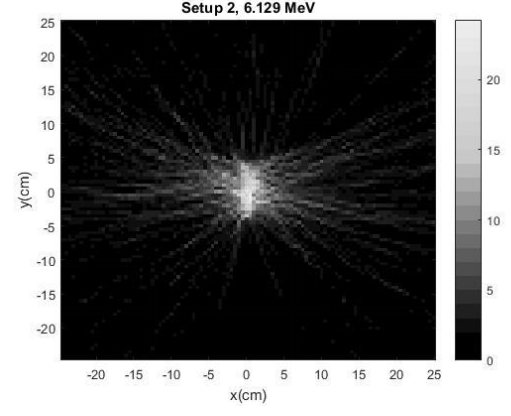
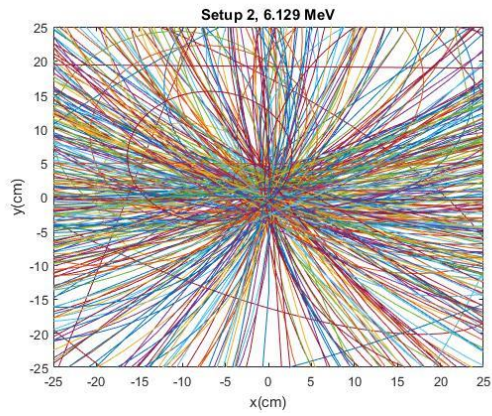
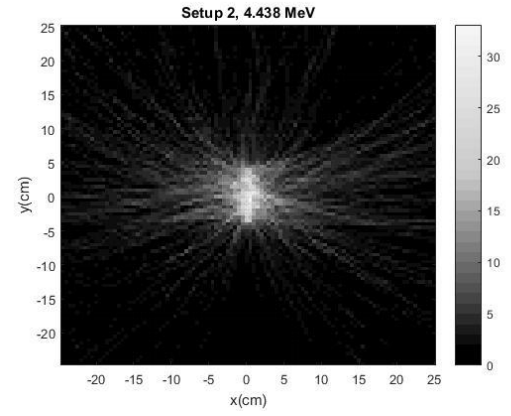
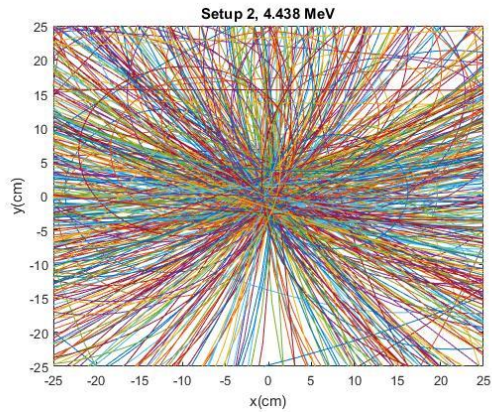
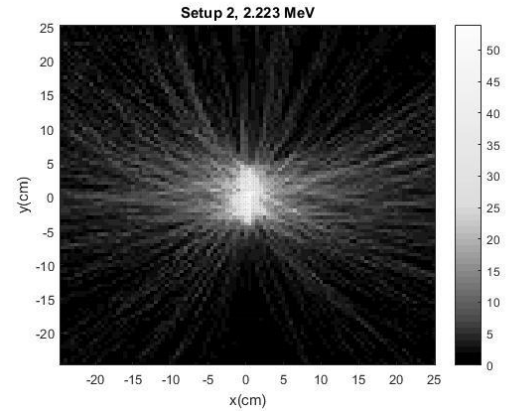
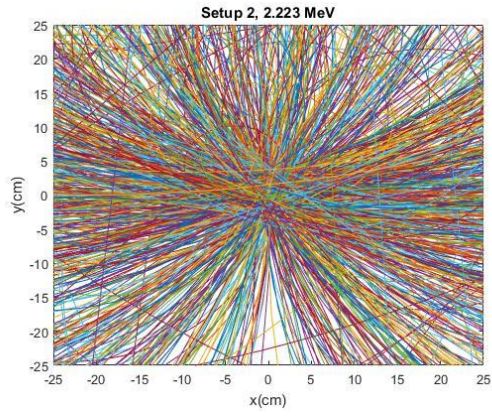


Figure 24: A plot of all the ellipses created with the written program for both tested setups and all four tested energies

Figure 23: All 8 created pixelated images using the given data

Figure 25 displays a Gaussian fit for the values of d at the middle of the image, while table 7 shows its parameters.

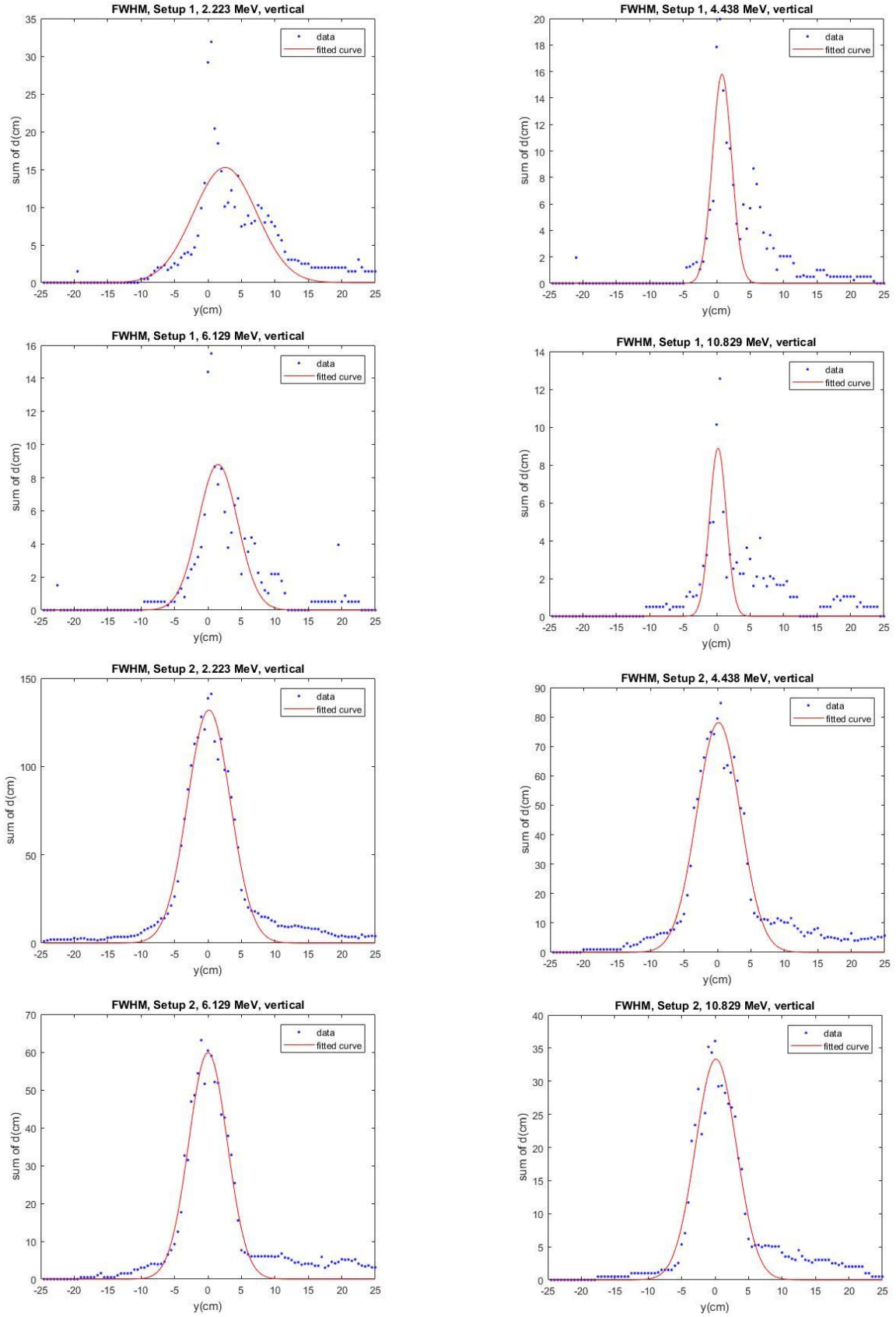


Figure 25: Vertical cuts of all 8 simulations including Gaussian fit

Gaussian formula:

$$f_{gauss}(x) = a_1 * e^{-\left(\frac{x-b_1}{c_1}\right)^2}$$

	E_{PG} (MeV)	a_1	b_1	c_1	FWHM(cm)	R^2
Setup 1	2.223	15.27	2.556	6.687	11,13	0.6853
	4.438	15.79	0.8068	2.024	3,370	0.7043
	6.129	8.798	1.497	4.171	6,945	0.7344
	10.829	8.892	0.2088	1.760	2,930	0.5853
Setup 2	2.223	131.9	0.148	4.358	7,256	0.9727
	4.438	78.09	0.1602	4.598	7,656	0.9460
	6.129	59.82	0.008244	4.070	6,777	0.9542
	10.829	33.34	0.1325	4.328	7,206	0.9383

Table 7: parameters of the Gaussian fit for every simulation and the accuracy R^2

5. Discussion

5.1 Optimal orientation

Using the code created with Matlab, the given datasets containing information about the PGs emitted by the gamma source were transformed. As a result, an image of the Compton cones could be made and a map showing the highest probable location of the PG source was created. However, there are a few remarks to be made.

First of all, looking at *figure 23*, it is immediately visible that setup 2(S2) has a higher efficiency when it comes to locating the PG source small number of data points leads to a large uncertainty. Reason for this inaccuracy is the fact that the randomness of the Monte Carlo simulation plays a bigger role with a smaller number of data. *Table 8* shows the efficiency to detect triple interaction events for both setups and the four energies.

	2.223 MeV	4.438 MeV	6.129 MeV	10.829 MeV
Setup 1	1.9%	1.1%	0.8%	0.5%
Setup 2	11%	5.9%	4.1%	2.4%

Table 8: Efficiency of both setups for all four tested energies

When comparing the images of setup 1(S1) to those of setup 2(S2), one can notice immediately that the images of setup 2 are more reliable when predicting the place of the PG source. All images of S2 show the highest probability of the source around the point $x = y = 0$, as it should. S1 does not have this clear central point, not even for the image that was formed with the largest dataset (the one with $E_{PG} = 2.223\text{MeV}$).

Another reason why S2 is more efficient than S1 is because of the difference in thickness of the detectors. Since the thickness of S1 is smaller, the probability of an interaction occurring is also smaller. Since for high energies, Compton scattering has a larger chance to occur in a forward direction, the probability to catch a consecutive interaction is higher for S2.

However, when looking at a gamma source sending

It would be interesting for a follow-up study to check if this difference is still this big if the amount of useable PGs would be larger, for example, taking 100,000 initial PGs instead of 10,000. The white dot(high probability) for setup 2 still has a fairly large error. There is no specific maximum. The maximum could become more distinguishable for both S1 and S2 if there was more data available. Another possible way to the data analysis and image reconstruction algorithms.

5.2 Code improvements

For the intersection-finding script, three($g = 3$) points evenly distributed in the range $t = [0, 2\pi)$ were taken. Using four or more points did not increase the number of conic sections found that crossed the Area of Interest, while taking only two or one point did decrease the found quantity. An example for S2 with an energy of $E_{PG} = 2.223\text{ MeV}$ is shown in *table 9*. However, for a lot of conic sections, no intersections were found. This is strange, since the origin of all PGs was known, meaning that most PGs should cross an Area of Interest containing the PG source. It seems unlikely that almost 25% of the ellipses do not come close to the

g	ellipses with intersections found
2	728/1014
3	767/1014
4	767/1014
5	767/1014

Table 9: A comparison showing that adding more than three initial guess points g does not increase the amount of ellipse intersections with the grid found

exact location of the PG source. This is probably due to inaccuracy in calculations. Since some ellipses had major axes of over 500, an inaccuracy could shift the ellipse.

The intersection-finding script has some parts that could be done better. Another element of the intersection-finding script could be improved. While finding intersection points and tracking the line through the grid, the code does not take into account the possibility that an ellipse can enter the grid for a second time. It moves on to the next conic section after finding the other crossing with the border.

It also does not track extremely small ellipses that are completely included in the Area of Interest. Since the tracking software only tracks curves that intersect with the border, this special case gets left out. However, this occurs rarely, so its influence on the image can be seen as negligible. For example, for S2, $E_{PG} = 2.223 \text{ MeV}$, only 8 out of 1014 ellipses met this requirement.

Another point of improvement is the calculation time of the program. Using a standard laptop, some datasets would take over a minute to make the transition from data to pixelated image. For larger datasets, this could escalate to extreme calculation times for bigger data sets. *Table 10* shows examples of computing times for all datasets. *Table 11* shows an example for the total time every part of the code took. Especially the command *sortrows* and the script *right* took a lot of time. There is a logical explanation for the long runtime of *right*. This script was a part of the *tracking* script and it was opened every time a right border of a pixel was crossed. Since *tracking* was coded in such a way that if a conic section had an intersection with the left border ($x = x_{min}$), it would take those coordinates as a starting point for the tracking. This meant that most of the time, *right* was the dominant direction for most of the tracking. The scripts *up* and *down* also take up a significant part of the total time. Since most curves did not cross the grid in a horizontal line, the upper and lower borders of pixels were also often crossed. This led to the frequent use of the *up*- and *down*-scripts.

The long wait can maybe be remedied by a more efficient tracking code. This code could for example use some other trick instead of the *sortrows* command. This could be another follow-up study.

Function Name	Calls	Total Time
matrixZ	1	76.757 s
tracking	1	64.260 s
right	57653	37.164 s
sortrows	89497	22.133 s
sortrowsParseInputs	89497	13.457 s
up	13523	9.056 s
down	13443	8.686 s
sortrowsParseInputs>directions2columns	85662	6.127 s
ellipseE	1	4.219 s
sortrows>sortBackToFront	89497	3.966 s
intersect	4	3.263 s
ellipse	1	2.706 s
sortrowsParseInputs>isAscendOrDescend	85662	1.876 s
hold	1781	1.620 s
crossright	84619	1.443 s
variablesE3	2	1.339 s
sortrowsParseInputs>parseNV	85662	1.290 s
crossup	72219	1.237 s
crossdown	72139	1.234 s
newplot	1784	1.191 s
left	1043	0.788 s
sortrowsParseInputs>legacyParseCOL	86429	0.613 s

Table 10: Calculation time for every component of the script

	E_{PG} (MeV)	# of PGs	Calculation Time (s)	Calculated PGs per second (s^{-1})
S1	2.223	1014	75.266	13,47
	4.438	588	37.265	15,78
	6.129	422	17.645	23,91
	10.829	236	12.544	18,81
S2	2.223	191	8.052	23,72
	4.438	112	5.453	20,539
	6.129	82	4.505	18,02
	10.829	51	3.399	15,00

Table 11: A comparison of calculation times for the entire script for all data

6. Conclusion

Just looking at the graphs presented in the **results** section shows that using the round side of the Ge detector creates an more accurate image of a Prompt Gamma source. If the detector would be used for proton therapy, setup 2 would be the best option to create an image of the irradiated tissue. The prime reason for this result is the higher efficiency in catching the PGs, which leads to an image which predicts the location of the PG source with higher efficiency.

The script written in MATLAB still has a lot of elements that could be improved. However, it is still an useful tool when it comes to comparing the effectiveness of various detectors.

References

1. How photons interact with matter. http://meroli.web.cern.ch/meroli/Lecture_photon_interaction.html. Accessed February 25, 2018.
2. PHYS 352 Photon Interactions Photon Interactions in Matter. <https://www.physics.queensu.ca/~phys352/lect17.pdf>. Accessed February 25, 2018.
3. Interactions of Photons with Matter The probability of an interaction per g cm⁻² of material traversed. Units of cm² g. https://ocw.mit.edu/courses/nuclear-engineering/22-55j-principles-of-radiation-interactions-fall-2004/lecture-notes/ener_depo_photon.pdf. Accessed April 3, 2018.
4. Thomson Scattering. <https://farside.ph.utexas.edu/teaching/jk1/lectures/node107.html>. Accessed April 3, 2018.
5. Blue Sky and Rayleigh Scattering. <http://hyperphysics.phy-astr.gsu.edu/hbase/atmos/blusky.html#c2>. Accessed April 3, 2018.
6. Johns THE, Cunningham JR. The Basic Interactions between Photons and Charged Particles with Matter Radiation Dosimetry I Thomson scattering (classical treatment) Thomson scattering (classical treatment) Compton scattering kinematics Compton scattering kinematics Compton scatte. :1-5.
7. Pajevic M. Photon Interaction. :2-4.
8. Abril I, Garcia-Molina R, de Vera P, Kyriakou I, Emfietzoglou D. *Inelastic Collisions of Energetic Protons in Biological Media*. Vol 65. Elsevier Inc.; 2013. doi:10.1016/B978-0-12-396455-7.00006-6.
9. Why is proton therapy a preferable option, and what is the Bragg peak? | Proton Therapy Today. <http://www.proton-therapy-today.com/what-is-proton-therapy/why-is-proton-therapy-a-preferable-option-and-what-is-the-bragg-peak/>. Accessed March 2, 2018.
10. Mohan R, Grosshans D. Proton therapy – Present and future ☆. *Adv Drug Deliv Rev*. 2016;109:26-44. doi:10.1016/j.addr.2016.11.006.
11. Jette D, Chen W. Creating a spread-out Bragg peak in proton beams. *Phys Med Biol*. 2011;56(11). doi:10.1088/0031-9155/56/11/N01.
12. Lee S, Brown GE. Hans Albrecht Bethe. 2 July 1906 -- 6 March 2005: Elected ForMemRS 1957. *Biogr Mem Fellows R Soc*. 2007;53(0):1-20. doi:10.1098/rsbm.2007.0018.
13. Narins PM. Scientific American, a division of Nature America, Inc. *Sci Am*. 1995;273(2):78-83.
14. Bichsel H. *Stochastics of Energy Loss and Biological Effects of Heavy Ions in Radiation Therapy*. Vol 65. Elsevier Inc.; 2013. doi:10.1016/B978-0-12-396455-7.00001-7.
15. Bernal-Rodríguez MA, Liendo JA. *Single Ionization of Liquid Water by Protons, Alpha Particles, and Carbon Nuclei: Comparative Analysis of the Continuum Distorted Wave Methodologies and Empirical Models*. Vol 65. Elsevier Inc.; 2013. doi:10.1016/B978-0-12-396455-7.00008-X.
16. Azizi M, Mowlavi A. Comparison of the Light Charged Particles on Scatter Radiation Dose in Thyroid Hadron Therapy. 2013:115-122.
17. Degiovanni A, Amaldi U. History of hadron therapy accelerators. *Phys Medica*. 2015;31(4):322-332. doi:10.1016/j.ejmp.2015.03.002.
18. McCleskey M, Kaye W, Mackin DS, Beddar S, He Z, Polf JC. Evaluation of a multistage CdZnTe Compton camera for prompt γ imaging for proton therapy. *Nucl Instruments Methods Phys Res Sect A Accel Spectrometers, Detect Assoc Equip*. 2015. doi:10.1016/j.nima.2015.02.030.
19. Knopf AC, Lomax A. In vivo proton range verification: A review. *Phys Med Biol*. 2013;58(15):131-160. doi:10.1088/0031-9155/58/15/R131.
20. Perali I, Celani A, Bombelli L, et al. Prompt gamma imaging of proton pencil beams at clinical dose rate. *Phys Med Biol*. 2014;59(19):5849-5871. doi:10.1088/0031-9155/59/19/5849.
21. Jan M-L, Lee M-W, Huang H-M. PSF reconstruction for Compton-based prompt gamma imaging. *Phys Med Biol*. 2017. doi:10.1088/1361-6560/aa9e74.
22. Park JH, Kim S, Kim CH, Seo H, Park S-H, Kim H-D. Development of a Compton Camera for

- Safeguards Applications in a Pyroprocessing Facility. *J Korean Phys Soc.* 2014;65(87). doi:10.3938/jkps.65.1360.
23. Sinclair L, Saull P, Hanna D, Seywerd H, MacLeod A, Boyle P. Silicon photomultiplier-based Compton Telescope for Safety and Security (SCoTSS). *IEEE Trans Nucl Sci.* 2014;61(5):2745-2752. doi:10.1109/TNS.2014.2356412.
 24. Schoene S, Enghardt W, Fiedler F, et al. An Image Reconstruction Framework and Camera Prototype Aimed for Compton Imaging for *In-vivo* Dosimetry of Therapeutic Ion Beams. *IEEE Trans Radiat Plasma Med Sci.* 2017;1(1):96-107. doi:10.1109/TNS.2016.2623220.
 25. MacKin D, Polf J, Peterson S, Beddar S. The effects of Doppler broadening and detector resolution on the performance of three-stage Compton cameras. *Med Phys.* 2013;40(1):1-12. doi:10.1118/1.4767756.
 26. Cone | Definition of Cone by Merriam-Webster. <https://www.merriam-webster.com/dictionary/cone>. Accessed March 27, 2018.
 27. Krimmer J, Dauvergne D, Létang JM, Testa. Prompt-gamma monitoring in hadrontherapy: A review. *Nucl Instruments Methods Phys Res Sect A Accel Spectrometers, Detect Assoc Equip.* 2018;878(August 2017):58-73. doi:10.1016/j.nima.2017.07.063.
 28. Downey DM, Sandy TA. Prompt gamma activation analysis of hydrogen. *J Chem Educ.* 1986;63(2):178. doi:10.1021/ed063p178.
 29. Kelleter L, Wrońska A, Besuglow J, et al. Spectroscopic study of prompt-gamma emission for range verification in proton therapy. *Phys Medica.* 2017;34:7-17. doi:10.1016/j.ejmp.2017.01.003.
 30. Saeed A, Nafee SS, Shaheen SA, et al. Calculating the ambient dose equivalent of fast neutrons using elemental composition of human body. *Appl Math Comput.* 2016;274:604-610. doi:10.1016/j.amc.2015.11.037.
 31. Chung C, Wei YY, Chen YY. Determination of Whole-body Nitrogen and Radiation Assessment Using In Vivo Prompt Gamma Activation Technique. *Appl Rad& t ht.* 1993;44(6):941-948. https://ac-els-cdn-com.proxy-ub.rug.nl/096980439390049G/1-s2.0-096980439390049G-main.pdf?_tid=2494c121-7d4c-406f-9ab3-cf4fc41bf153&acdnat=1520099610_865a2de77562a701189419a27f5a7f5c. Accessed March 3, 2018.
 32. Singh VP, Badiger NM. Photon interaction with semiconductor and scintillation detectors. *Nucl Sci Tech.* 2016;27(3):1-12. doi:10.1007/s41365-016-0076-8.
 33. Lee W, Bolotnikov A, Lee T, et al. Mini Compton Camera Based on an Array of Virtual Frisch-Grid CdZnTe Detectors. *IEEE Trans Nucl Sci.* 2016;63(1):259-265. doi:10.1109/TNS.2015.2514120.
 34. Los Alamos National Laboratory: MCNP Home Page. <https://mcnp.lanl.gov/>. Accessed March 2, 2018.
 35. Dendooven P, Angelique J. COMPTON CAMERA IN PROTON. 2017.
 36. GeGI Info Sheet - PHDS Co. - Germanium Gamma Ray Imaging Detectors. <http://phdsco.com/products/gegi/gegi-info-sheet>. Accessed March 5, 2018.
 37. Otto SR. [5] *An Introduction to Programming and Numerical Methods In Matlab*. Vol 1542.; 2015. doi:10.1017/CBO9781107415324.004.
 38. Lojaco X. Image reconstruction for Compton camera with application to hadrontherapy. 2014:200.