Faculteit Industriële Ingenieurswetenschappen

master in de industriële wetenschappen: elektronica-ICT

Masterthesis

Position determination of a gamma ray point source using a single layer Compton camera

Anouk Michiels

Scriptie ingediend tot het behalen van de graad van master in de industriële wetenschappen: elektronica-ICT

PROMOTOR :
Prof. dr. ir. Jan GENOE

PROMOTOR :
Prof. dr. Wouter SCHROEYERS

BEGELEIDER :
drs. ing. Mattias SIMONS

Gezamenlijke opleiding UHasselt en KU Leuven

Universiteit Hasselt | Campus Diepenbeek | Faculteit Industriële Ingenieurswetenschappen | Agoralaan Gebouw H - Gebouw B | BE 3590 Diepenbeek

Universiteit Hasselt | Campus Diepenbeek | Agoralaan Gebouw D | BE 3590 Diepenbeek
Universiteit Hasselt | Campus Hasselt | Martelarenlaan 42 | BE 3500 Hasselt

2022
2023

Faculteit Industriële Ingenieurswetenschappen

master in de industriële wetenschappen: elektronica-ICT

Masterthesis

Position determination of a gamma ray point source using a single layer Compton camera

Anouk Michiels

Scriptie ingediend tot het behalen van de graad van master in de industriële wetenschappen: elektronica-ICT

PROMOTOR :

Prof. dr. ir. Jan GENOE

PROMOTOR :

Prof. dr. Wouter SCHROEYERS

BEGELEIDER :

drs. ing. Mattias SIMONS

UHASSELT KU LEUVEN

Preface

Finally.

Now that I am finishing my master's thesis, I want to start by thanking my mother and father for supporting me throughout all those years.

When I started this study almost eight years ago, I did not know which specialty I wanted to pursue. I presumed I did not want to do civil engineering or anything that involved "too much computer stuff". However, after the first few programming lessons, I quickly found my favourite subject. This enthusiasm was magnified by the boy — man by now — sitting next to me. Laurens, that first week, you started by teaching me how to zip a folder and just last week you helped me out with some LaTeXproblems. I can always count on you when I need practical help or mental support. Thank you for being there.

Another thank you goes to the women in my life, especially my two sisters and my sister-in-law who had and have successful careers in academia and technology. Thank you for showing me how it is done and for honestly advising me that (even though the world is starting to get it right) yes, you do have a better chance at being taken seriously if you go for the more technical subject. I also want to thank Mevrouw Mentens, who — probably without knowing it — is a role model for every female student in this male-dominated field.

After getting my assignment of choice, not yet knowing the supervisors involved in this subject, I quickly realised I drew the long straw. To all my supervisors, thank you for your flexible attitude that allowed me to do my research on my own pace. Meneer Genoe, thank you for all your technical guidance and for entertaining many of my "why's" regarding academic guidelines. I also greatly appreciated you making time for frequent follow-ups, often providing reassurance when I wasn't too confident in my own work. Meneer Schroeyers, thank you for your effort to squeeze in our meetings between many other obligations and your forever positive attitude. Mattias, thank you for being such an accessible advisor. Your flexible attitude and fast email responses made my work a whole lot easier.

I also had the privilege to have a lot of academic, technical and English knowledge at my fingertips. Thank you to my father and *brusjes*[1], who joined several debates about correct translations for words and the best ways to clarify illustrations. Thank you, Daan and Annelies, for letting me pick your math brains when mine was confused. Thank you, Laura and Karolien, for letting me vent when necessary and for proofreading parts of my

[1] broers en zussen = brothers and sisters

thesis. I've looked up to all three of you since I was a little kid and twenty years later, I still do.

Mevrouw Bovin, thank you for always being there. Without your unwavering support, I would not have made it to this point. I also want to thank all the lecturers who supported me during all these years and who did more than their jobs to help me out. Thank you for having my back in this longer-than-expected journey.

Nico, thank you for being my rock during all these years and during this thesis. Thank you, for your immense patience with my frustration when things didn't go as I planned. Which they did. Often. Thank you for reminding me to take breaks and go sleep when I was too hyper-focused on my work to remember. Thank you for taking care of me and for always being there. Thank you for being you.

- Anouk

Table of Contents

List of Tables

List of Figures

List of symbols

Chapter 2	
E_γ [keV] = E_0 [keV]	energy of the incoming γ ray
E'_γ [keV]= E_2 [keV]	energy of the scattered γ ray
E_e [keV] = E_1 [keV]	energy of the recoil electron
θ [°]	scattering angle, opening angle, cone angle
$\vec{u}$	normalised scatter axis, cone axis
Chapter 4	
ϕ [°]	angle obtained by rotating around horizontal axis
τ [°]	angle obtained by rotating around vertical axis
A [MBq]	activity of a source
f	number of frames
uf	number of usable frames
R_e	energy range, how much the total energy of a cluster may deviate from the ideal energy E_γ
Chapter 7	
n	number of measurements
$P_m = [\phi_m, \tau_m]$	measured intersection point, defined by the angles ϕ_m [°] and τ_m [°]
$P_c = [\phi_c, \tau_c]$	calculated intersection point, defined by the angles ϕ_c [°] and τ_c [°]
Chapter 8	
a_S [°]	resolution of the discretised unit sphere
S	2D matrix that represents the discretised unit sphere
σ [°]	standard deviation used to create a single back-projection
δ_g	the Gaussian difference between a cone and the points of the discretised unit sphere
ε_a [°]	angular error, angular difference between P_m and P_c
$\delta_{m,c}$	difference between a cone and the points of the discretised unit sphere, expressed as a value between $\ln(v_c)$ and 0, used to calculate a single 2D map before multiplication of the 2D maps
v_c	cut-off value, used in making a single back-projection for multiplication
Chapter 9	
$\overline{\varepsilon_a}$ [°]	mean of a list of angular errors
RMS(ε_a) [°]	root-mean-square of list of angular errors
FWHM [keV]	full width at half maximum, used to determine the width of the absorption peak
l_t	total length
l_s	slice length
m	number of rows, number of values for independent variable
s	number of shuffles

$uf\%$ [%]	percentage of frames that are considered usable frames
uf_h $\left[\frac{frames}{h \cdot MBq}\right]$	usable frames per hour per MBq
t_{100} [h]	time necessary to gather 100 usable frames for a source with A = 1 MBq
std [°]	standard deviation used to describe a Gaussian distribution
SEM [°]	standard error on the mean

Abstract

A present-day challenge is nuclear decommissioning. Before actual decommissioning of a nuclear facility can begin, human operators locate and characterise the sources, hot spots and contaminated areas by entering the facility and manually performing measurements. A strict time limit for these measurements ensures the health of the operators, but leads to less accurate or incomplete measurements. A mobile robot that maps the sources using a single layer Compton camera, can be a viable alternative.

This thesis researches how to use a 1 mm thick CdTe single layer Compton camera to detect the position of a γ-ray point source at a given distance from the detector. Back-projection of the Compton cones, which are reconstructed using the data from the camera, determines the two angles necessary to locate the point source. Two back-projection algorithms are implemented and compared: addition and multiplication.

The implemented algorithms have similar results: they both show an exponential descent for the angular error in function of the measurement time. Typically, addition is 6% more accurate compared to multiplication, but that percentage is larger for shorter measurement times and shrinks to zero for longer measurement times. For a γ-ray point source with an activity of 1 MBq, the angular error for addition is 2.90° at a measurement time of 1 h and 1.77° at a measurement time of 4 h. Those results can compete with the current state-of-the-art.

Abstract in het Nederlands

Een hedendaagse uitdaging is de ontmanteling van nucleaire faciliteiten. Vooraleer de daadwerkelijke ontmanteling van een dergelijke faciliteit kan beginnen, lokaliseren menselijke operatoren bronnen, hot spots en gecontamineerde gebieden door de faciliteit te betreden en handmatig metingen uit te voeren. Een strikte tijdslimiet is noodzakelijk om de gezondheid van de operatoren te waarborgen, maar leidt tot minder nauwkeurige of onvolledige metingen. Een mogelijk alternatief is een mobiele robot die de bronnen in kaart brengt met behulp van een eenlagige Compton-camera.

Deze thesis onderzoekt hoe een 1 mm dikke CdTe eenlagige Compton-camera gebruikt kan worden om de positie van een γ-puntbron te bepalen die zich bevindt op een gegeven afstand van de detector. Projectie van de Compton-kegels, die worden gereconstrueerd op basis van de data van de camera, bepaalt de twee hoeken die nodig zijn om de puntbron te lokaliseren. Twee projectie-algoritmen worden geïmplementeerd en vergeleken: optelling en vermenigvuldiging.

De twee algoritmen hebben vergelijkbare resultaten: ze tonen beide een exponentiële daling voor de hoekfout in functie van de meettijd. Optelling is typisch 6% accurater dan vermenigvuldiging, maar dat percentage stijgt voor kortere meettijden en daalt tot nul voor langere meettijden. Voor een γ-puntbron met een activiteit van 1 MBq is de hoekfout voor optelling 2,90° bij een meettijd van 1 u en 1,77° bij een meettijd van 4 u. Deze resultaten zijn in lijn met de huidige stand van de techniek.

Chapter 1

Introduction

A present-day challenge is nuclear decommissioning. Before actual decommissioning of a nuclear facility can begin, the sources, hot spots and contaminated areas have to be located and characterised. Currently, human operators carry out this task by entering the facility and manually performing the measurements [1].

The most significant disadvantage is that the operators are exposed to radiation. To keep the radiation exposure to a reasonable minimum, there is a strict time limit in which the measurements have to be performed. The reasonable minimum is determined according to the ALARA (as low as reasonably achievable) principle [2]. This ensures the health of the workers, but leads to less accurate or incomplete measurements. Using a mobile robot to perform these measurements, can be a viable alternative [3].

One of the techniques the robot can use to map the sources is a compact Compton imaging system. A Compton camera uses incoming gamma rays to reconstruct Compton scattering events. These events can then be used to estimate the position of the source. A traditional Compton camera consists of two layers [4]. However, there is a second type of Compton camera: a single layer Compton camera [5]. This type of camera has a lower efficiency, but is smaller and lighter than a traditional Compton camera and thus easier to mount on a mobile robot. This paper will research the possibility of using a single layer Compton camera to locate a γ-ray source.

This thesis builds upon the research done in [3]. Chapter 2 starts with a literature study about the interactions of γ radiation with matter and how Compton scattering is the basis of the workings of the traditional and single layer Compton camera. Next, chapter 3 describes a study of the research performed in [3] and the specific γ camera used in that research.

This is followed by chapter 4, which explains the measurement setup and the used software. Subsequently, in chapter 5, reverse engineering of pieces of code already written in [3] is performed in order to start from a structured framework, which can be used for Compton processing. Chapter 6 gives an overview of the used conventions and chapter 7 analyses the gathered data as a whole.

Chapter 8 explains the creation of back-projections in order to determine the position of the point source. Different algorithms were implemented: some are based on addition

of back-projections while others are based on multiplication of back-projections. Next, chapter 9 discusses the results of the different implementations with different parameters. This is followed by a comparison between the different implementations and the current state-of-the-art. The thesis ends with a concise conclusion and outlook in chapter 10.

Chapter 2

Compton scattering

2.1 Nuclear decay

Radioactive materials emit radiation due to nuclear decay. In the process of nuclear decay, an unstable atomic nucleus loses energy by radiation. In α decay the nucleus emits an α particle, which is a ${}^{2}_{4}He^{2+}$. In β decay, a β particle is emitted. A β particle is a high-energy electron or positron emitted by the atomic nucleus. α decay or β decay can be followed by γ decay which involves the emitting of a γ ray, which is a photon [6].

If α or β particles end up in the body they do significantly more damage than a γ particle would do, but they only travel a small distance in the air. An α particle can travel about 2.5 cm in the air and cannot penetrate the skin. β particles can travel several meters in the air and can penetrate almost 1.2 cm of skin and body, but they can be shielded with less than 2.5 cm of a material such as plastic. γ rays can travel tens of meters and can easily penetrate the body. Shielding of γ rays requires thick and dense material, such as concrete or lead [7].

This thesis focuses on the detection of sources of γ radiation.

2.2 Interaction of γ radiation with matter

γ radiation can interact with matter in different ways, depending on the energy of the γ radiation and the atomic number of the absorber. At low energy, **photoelectric absorption** occurs the most, at mid-energy range **Compton scattering** and at high energy, **pair production**. Figure 2.1 visualises this. On the left curve, the photoelectric effect and Compton effect are equally dominant, while on the right curve, the Compton effect and pair production are equally dominant [8].

2.2.1 Photoelectric absorption

In photoelectric absorption, a γ ray photon interacts with one of the bound electrons of an atom. The kinetic energy of the electron E_e is the difference between the γ ray energy E_γ and the energy necessary to separate the electron from its shell E_b. Figure 2.2 displays the mechanism [8].

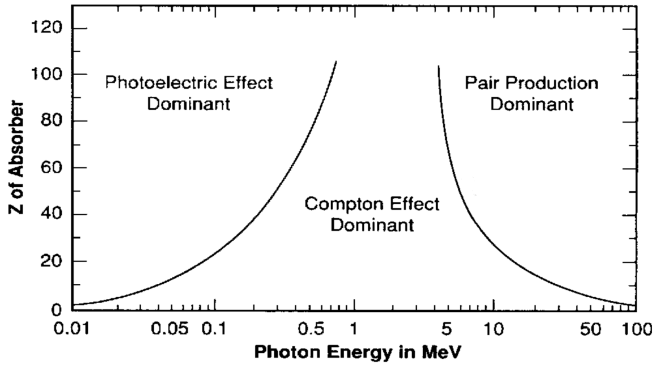


Figure 2.1: The dominance of the three mayor γ ray interactions relative to the atomic number of the absorber and the energy of the incoming γ ray, adapted from [9]

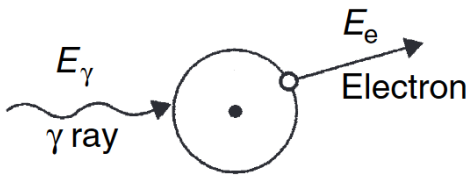


Figure 2.2: The mechanism of photoelectric absorption [8]

2.2.2 Compton scattering

Compton scattering is a direct interaction of a γ ray photon with a loosely bound electron. During this interaction, the γ ray collides with an electron. A part of the energy of the original γ ray is given to the recoil electron and is that electrons kinetic energy. The rest of the energy manifests as a scattered γ ray γ' with a scattering angle θ. The binding energy of the electron is ignored because it is only a few eV and thus negligible compared to the energy of the incoming γ ray γ, which is a few keV. Figure 2.3 illustrates this interaction [8].

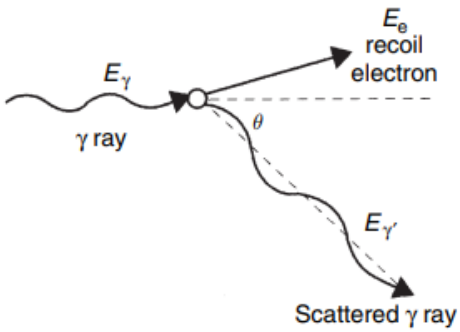


Figure 2.3: The mechanism of Compton scattering [8]

Compton equation

The energy of the recoil electron is calculated using equation 2.1 [4].

$$E_e = E_\gamma - E_{\gamma'} \qquad \text{with} \begin{cases} E_e \text{ [keV] is the energy of the recoil electron} \\ E_\gamma \text{ [keV] is the energy of the incoming gamma ray} \\ E_{\gamma'} \text{ [keV] is the energy of the scattered gamma ray} \end{cases} \tag{2.1}$$

The scattering angle is calculated using equation equation 2.2 [4].

$$\cos(\theta) = 1 - m_e c^2 \frac{E_e}{E_\gamma (E_\gamma - E_e)} \qquad \text{with} \begin{cases} \theta \text{ is the scattering angle} \\ m_e c^2 = 511 \text{ keV} \end{cases} \tag{2.2}$$

The relation between the scattering angle θ and the three different energies is given by equation 2.3, which combines equations 2.1 and 2.2 and is referred to as the Compton equation [4].

$$\begin{cases} \cos(\theta) = 1 - m_e c^2 \frac{E_e}{E_\gamma (E_\gamma - E_e)} \\ E_\gamma = E_e + E_{\gamma'} \end{cases} \tag{2.3}$$

2.2.3 Pair production

In pair production, the γ ray photon interacts with the atom as a whole. The γ ray photon converts into a electron-positron pair. This means that pair production can only occur when the γ ray photon has an energy equivalent to at least the combined rest mass of the two particles (electron and positron), which is $2 \cdot 511$ keV $= 1022$ keV. In practice, pair production occurs at energies higher than 1022 keV. The two created particles share the excess energy of the γ ray photon equally. The positron loses kinetic energy very fast in matter until it reaches a kinetic energy close to zero. Because positrons are anti-particles to electrons and the slowed positron will eventually be near an electron, they exist shortly as a positronium. Then the process of annihilation occurs and both the positron and electron disappear and two photons are produced, each with an energy equal to the electron mass of 511 keV. Because that whole sequence is likely to happen within 1 ns of creation of the pair, it can be considered instantaneous in practice. Figure 2.4 visualises the sequence of events.

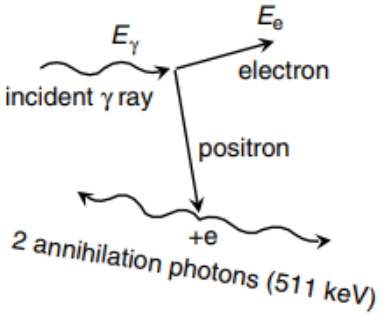


Figure 2.4: The mechanism of pair production [8]

2.3 Compton camera

The Compton equation can be used to reconstruct Compton events given that the energies and corresponding positions are known. This concept is the basis of the workings of a Compton camera. The figures of this section use different symbols than the ones established before: $E_0 = E_\gamma$, $E_1 = E_e$ and $E_2 = E_\gamma$'.

2.3.1 Traditional Compton camera

A traditional Compton camera relies on the principle of Compton scattering and consists of two detectors. The scattering of the primary γ ray (first interaction) occurs in the *scattering detector*, which records the position and energy of the recoil electron. The scattered photon continues towards the *absorption detector*. This second detector absorbs the scattered photon (second interaction) and records its energy and position. The Compton scattering events of the incoming γ ray can then be reconstructed using the collected data and the Compton equation (eq. 2.3) [4].

Figure 2.5 visualises how possible directions of the scattered γ ray can be presented as the surface of a cone. The energies of the events determine the opening angle of the cone θ and the positions of the events determine its axis of revolution. The axis of revolution is also called the scatter axis or the cone axis and its normalised version is referred to as $\vec{u}$.

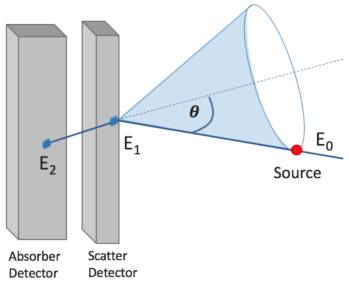


Figure 2.5: Visualisation of the possible directions of the original γ ray as the surface of a cone [4]

With multiple Compton interactions recorded, multiple cones can be constructed. The intersection of the surfaces of the cones determines the location of the source. Conic curves can be created using the intersection of the surfaces of the cones with a projection plane. This results in a back-projected image of the cones. Figure 2.6 illustrates this. Instead of choosing a plane as the projection surface, it is also possible to use a sphere. Section 8.1.1 discusses this option.

2.3.2 Single layer Compton camera

A single layer Compton camera, useful because of its size and portability, has the disadvantage of having only one detector. Although the scattering of the primary γ ray and the absorption of the scattered photon can occur in a single detector [5], two difficulties arise. First, there is no obvious depth difference between the two interactions, which means it is more difficult to get an accurate cone axis. Second, while two interactions still take place,

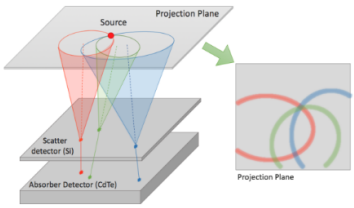


Figure 2.6: Back-projection onto a plane of multiple reconstructions using a traditional Compton camera [4]

it is usually not possible to determine which interaction represents the scattering of the primary γ ray and which interaction represents the absorption of the scattered photon. This means there are two cones possible cones, but it is not possible to determine which is the correct one. Figure 2.7 illustrates Compton events in a single layer Compton camera with only the correct cones (and their back-projections) displayed.

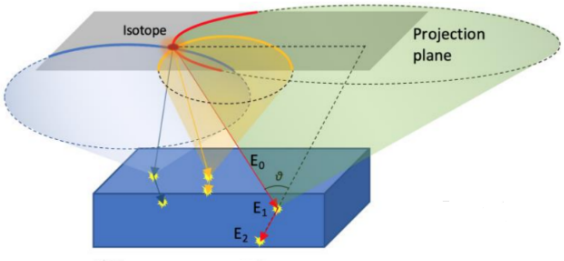


Figure 2.7: Back-projection onto a plane of multiple reconstructions from a single layer Compton camera [5]

Section 3.4 explains the determination of the estimated interaction depths. The problem of not being able to determine which cone is the correct one cannot be solved without using advanced techniques. Therefore, both cones will be used for further calculations.

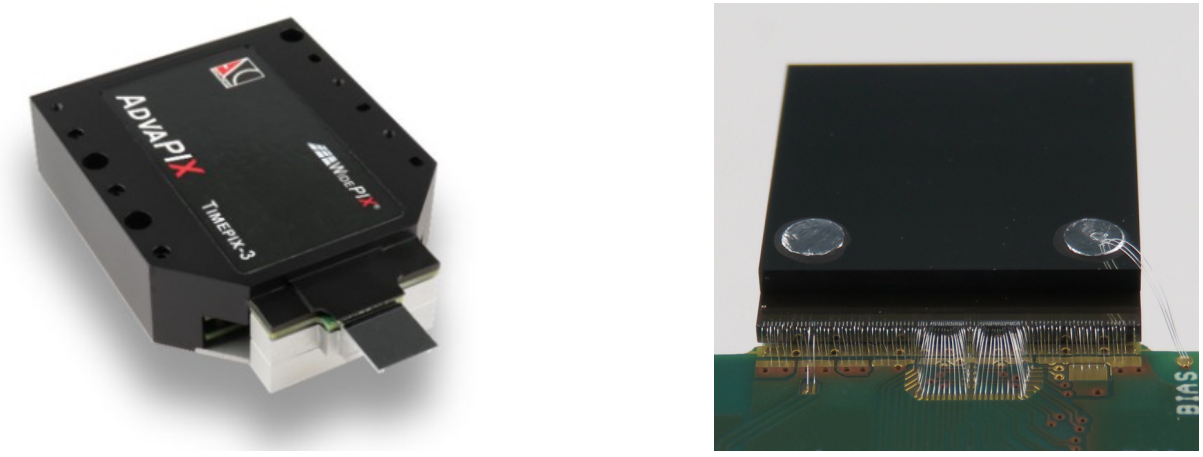
Chapter 3

Gamma camera

The exploration of the workings of the used Compton camera and the selected settings and parameters were part of a previous Master's thesis [3] and are therefore discussed separately in this chapter. No new decisions concerning the settings and parameters were made. This chapter exists solely to explain the workings of the camera and the previous research in order to lay a foundation for the rest of the thesis.

3.1 AdvaPIX TPX3

The measurements were done with a single layer Compton camera based on Timepix3 technology: the AdvaPIX TPX3, model B07-W0049 with a 1000 μm CdTe detector [10]. It is a hybrid pixel detector with a read-out chip that can record time-of-arrival (ToA) and time-over-threshold (ToT) simultaneously in each pixel. The detector consists of a grid of 256 by 256 pixels. Each pixel has a area of 55x55 μm [11]. Figure 3.1 shows the detector.



(a) Outside of the AdvaPIX TPX3 detector [10]

(b) Inside of the AdvaPIX TPX3 detector

Figure 3.1: AdvaPIX TPX3 detector

3.2 Hybrid pixel detector

Hybrid pixel detectors are radiation detectors that consist of an array of semiconductor technology diodes and their associated electronics. The semiconductor sensor and the ASIC (application specific integrated circuit) are manufactured separately and afterwards bump-bonded [12]. Bump-bonding is a technique used when a large number of connections are needed between the detector layer and the electronics that are as dense as the pixels themselves. In this case there are 256 by 256 pixels that have to be connected with their associated pixel channels with a pixel pitch of only 55 μm. Bump-bonding means the two layers are connected by a metal droplet. Solder bumps are placed on the die, the die is flipped over and the solder bumps are aligned with the contact pads on the substrate. Placement in a furnace ensures that re-flowing of the solder ball takes place which establishes the bonding between the die and the substrate [13]. Figure 3.2 illustrates bump-bonding.

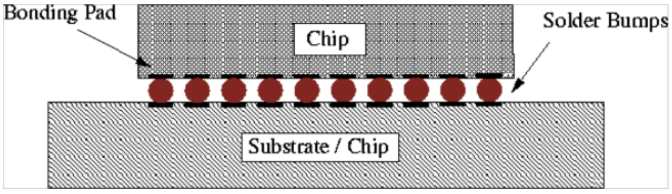
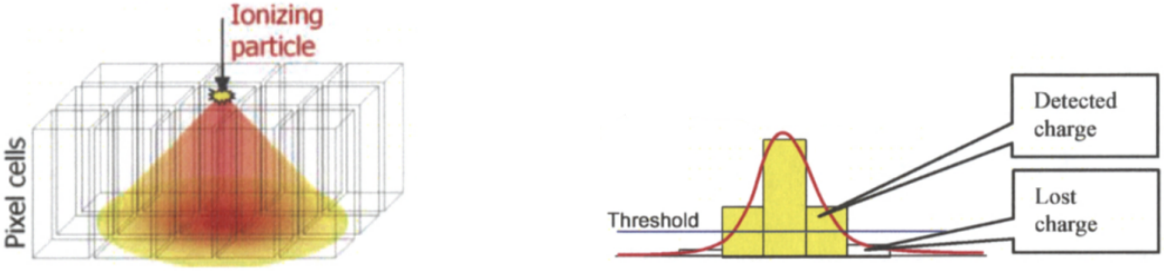


Figure 3.2: Bump-bonding [13]

3.2.1 Charge sharing

In a hybrid pixel detector, charge sharing occurs because the pixels are small and very close to one another. Under the influence of an electrical field, the charge carriers drift towards the corresponding electrode [14]. During this collection process, the charge carriers are spread out due to repulsion and diffusion. Repulsion means the charge carriers going to the same electrode have the same charge, so they repel each other and spread out. Diffusion means the charges do not exactly follow the electric field lines, but diffuse as a result of the random thermal motion in the crystal lattice [15]. This leads to the charge being spread out across several neighbouring pixels, thus forming a cluster of pixels. Figure 3.3a visualises the principle of charge sharing. For each pixel, the collected charge is compared to a threshold level and only if it is higher, the event is registered. This means that a part of the energy is lost due to not reaching the threshold in certain pixels [16]. That is illustrated in 3.3b.



(a) Charge sharing

(b) Lost charge

Figure 3.3: Charge sharing and lost charge as a result [16]

3.3 Signal formation

For each pixel the charge carriers induce current. This current is converted into voltage pulses in the nearest pixel electrode. This leads to a triangular pulse as illustrated in Figure 3.4 [12]. The different parameters that can be found in that figure are explained below [10], [12].

THL = threshold level
The threshold level is the minimum level of energy a pixel must have in order to be registered as a hit. The threshold level is given in keV.

ToA = time-of-arrival
The ToA is the time from the start of the exposure until the signal crosses the threshold level. It is expressed in the number of periods of a the base clock. The base clock has a frequency of 40 MHz.

fToA = fast-time-of-arrival
The fToA is the time between the signal crossing the threshold and the next base clock signal. An additional clock with a frequency of 640 MHz is used for this purpose. The more precise arrival time can therefore be calculated by using equation 3.1.

$$\text{arrival time [ns]} = \text{ToA} \cdot 25 \text{ ns} - \text{FToA} \cdot 1.5625 \text{ ns} \tag{3.1}$$

ToT = time-over-threshold
The ToT is the time the signal is above the threshold level. For this registration only the base clock is used, so it has an accuracy of 25 ns.

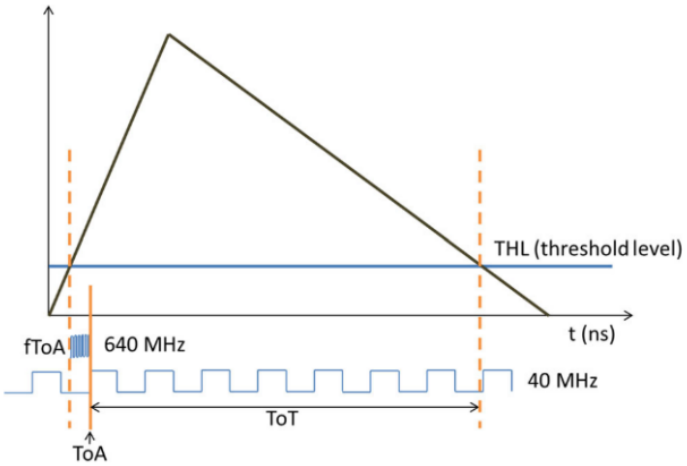


Figure 3.4: Signal formation with indication of different time parameters[12]

3.3.1 Time walk effect

The amplifier in the ASIC has a fixed signal rise time of ∼25 ns. The height of the signal is linearly correlated to the charge deposited in the pixel. This leads to the signal crossing the threshold at a different time for different energies [12], as illustrated in figure 3.5. An event with a lower energy and therefore a lower end value, crosses the threshold at a

later time. This is what is known as the time walk effect and is something that has to be taken into account when determining the correct time stamps. Equation 3.2 calculates the corrected ToA.

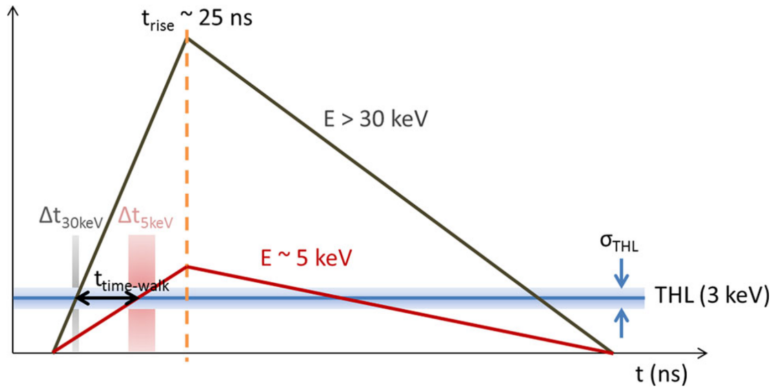


Figure 3.5: Time walk effect [12]

$$\text{ToA} = \text{ToA}_{measured} - \Delta\text{T} \qquad \text{with } \Delta\text{T the time walk} \tag{3.2}$$

In order to make a good estimate for the needed time-walk correction, a per-pixel calibration would be the ideal scenario. Because this was not possible due to restrictions in laboratory availability, a general time-walk correction was determined. The method discussed in [17] was used. That method results in equation 3.3 for the calculation of the time-walk effect [17].

$$\Delta\text{T [ns]} = \frac{c}{(E - E_0)^d} \qquad \text{with} \begin{cases} E \text{ [keV] is the energy of the pixel} \\ E_0 \text{ [keV] is the threshold energy} \\ \text{calibration parameter c } \left[\frac{ns}{\text{keV}^d}\right] \\ \text{calibration parameter d} \end{cases} \tag{3.3}$$

The results from [3] determine that c = 57179 $\frac{ns}{\text{keV}^d}$, d = 3.535 and E_0 = 5.00 keV.

3.3.2 Relation between ToT and energy

In order to accurately describe the relation between ToT and deposited energy, a per-pixel calibration of the device is necessary. This calibration does not have to be done by the user. It comes pre-installed with the device. To give an idea of what that relation looks like, figure 3.6 visualises the relation for one individual pixel. The relation is non-linear for energies below ~20 keV and linear for higher energies. The parameters a, b, c and t are the pre-installed values. This means the parameters can be viewed as an array of 65536 by 4. Equation 3.4 is the corresponding equation [18], [19].

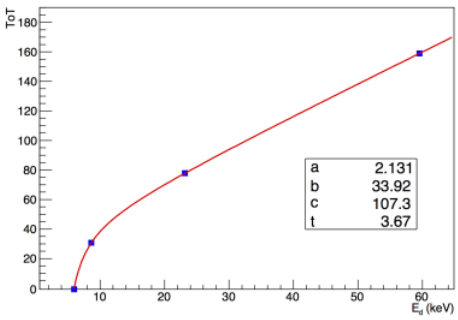


Figure 3.6: Calibration for an individual pixel [19]

$$\text{ToT [ns]} = a \cdot E_d + b - \frac{c}{E_d - t} \qquad \text{with} \begin{cases} E_d \text{ [keV] is the deposited energy} \\ \text{calibration parameter a } [\frac{\text{ns}}{\text{keV}}] \\ \text{calibration parameter b [ns]} \\ \text{calibration parameter c [ns} \cdot \text{keV]} \\ \text{calibration parameter t [keV]} \end{cases} \tag{3.4}$$

3.4 Depth determination

When using a camera with only a single layer, the depth cannot be measured directly and has to be constructed using other information available. The method explained in [20] was followed. Figure 3.7 shows the different time stamps. $t_{interaction}$ is the time of the interaction. t_{drift} is the delay due to the charge carrier drift. $t_{induction}$ is the time the pulse shaping starts. It starts a bit before the charge carriers actually arrive due to charge induction. t_{offset} is the time between the start of the pulse shaping and the time the hit is assigned. $t_{time-walk}$ is ΔT as mentioned in section 3.3.1. $t_{measurement}$ is the time the interaction is actually measured. As the figure illustrates there are delays due to charge carrier drift en pixel electronics behaviour. As illustrated in figure 3.7, t_{drift} is not exactly the same as the difference between $t_{induction}$ and $t_{interaction}$, but the difference is deemed negligible [20].

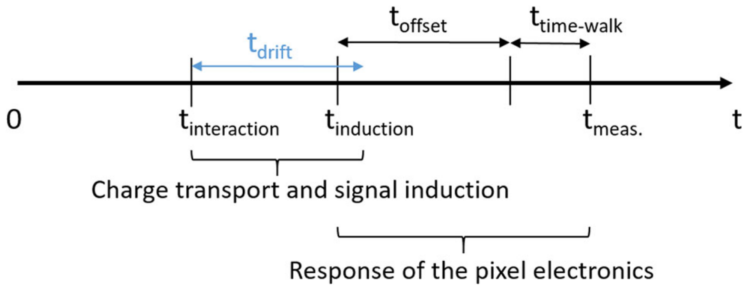


Figure 3.7: Illustration of the different time stamps [20]

The drift time measured in pixel i is the difference between the pixel's time stamp t_i and the minimal time within the set of pixels forming a track $t_{min,track}$. Because the depth depends on the drift time, the important parameter for depth determination is $t_{induction}$.

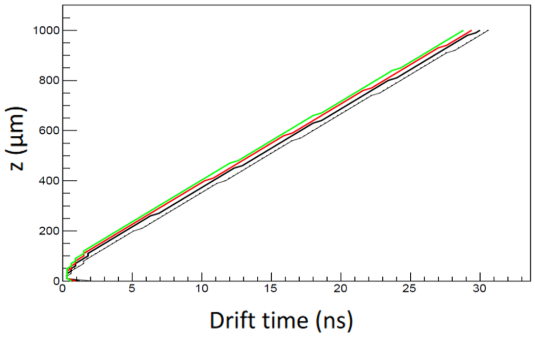
Since $t_{induction}$ depends on the interaction depth and the deposited energy, 2D lookup tables were calculated $z_{look-up}$ ($t_{induction}$, E_{dep}). The equation for the estimated depth is equation 3.5 [20].

$$z_{est} = z_{0,look-up}(t_i - t_{min,track},\ E_{measurement}) \tag{3.5}$$

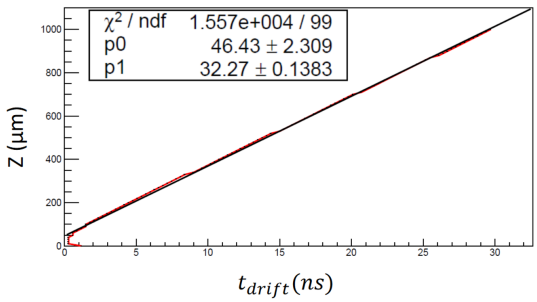
The look-up tables calculated in [3] for four different energy bins are displayed in figure 3.8a. It illustrates that pixels with a higher energy have a lower drift time for the same interaction height. Because all the look-up tables are close to one another, it was decided in [3] to do a linear approximation of one energy bin: 75-100 keV. Figure 3.8b visualises the linear approximation that is used to calculate the depth based on the difference in time-of-arrival. Equation 3.6a is used to calculate the depth for pixels with an energy above 30 keV. For pixels with energies below 30 keV a time-walk-correction is applied, which leads to equation 3.6b. The extra values used in equation 3.6b are those from section 3.3.1.

$$z = 46.43 + 32.27 \cdot \Delta ToA \tag{3.6a}$$

$$z = 46.43 + 32.27 \cdot \left(\Delta ToA - \frac{57179}{(E+5)^{3.535}}\right) \tag{3.6b}$$



(a) Look-up tables for 0-25 keV (grey), 25-50 keV (black), 50-75 keV(red) and 75-100 keV (green)



(b) Linear approximation of the look-up table for the energy bin of 75-100 keV

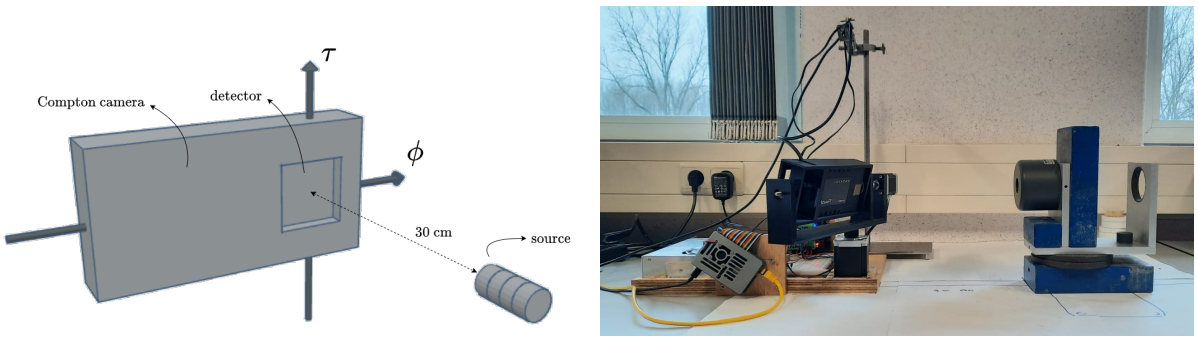
Figure 3.8: Look-up tables [3]

Chapter 4

Measuring setup and software

4.1 Measuring setup

The point source is located at a distance of 30 cm from the centre of the active part of the detector. The detector rotates around the ϕ and τ axes to gather measurements for different angles. The detector is connected to a laptop that uses the PIXetPro software [21] to capture and save all the measurement data. Figure 4.1 illustrates the setup.



(a) Illustration of the measurement setup

(b) Photo of the measurement setup

Figure 4.1: Measurement setup

4.1.1 Used source

A ^{137}Cs point source will be assumed. In practice, it will be a series of four point sources in a row. Table 4.1 lists the characteristics of the used sources. These activities to a total activity A = 14.15 MBq. Figure 4.2a displays the sources. The sources are stacked into wooden holders that fit in the lead cylinder as displayed in figure 4.2b. That lead cylinder can also be seen figure 4.1b.

Table 4.1: Characteristics of the four point sources

Source	Element	Activity	Uncertainty	y-impurities
1	^{137}Cs	3.52 MBq	5%	0.1%
2	^{137}Cs	3.55 MBq	5%	0.1%
3	^{137}Cs	3.52 MBq	5%	0.1%
4	^{137}Cs	3.56 MBq	5%	0.1%

(a) Four sources

(b) The sources in the wooden holders and the lead cylinder

Figure 4.2: Sources

4.1.2 Used gamma camera

The measurements are done with a single layer Compton camera based on Timepix3 technology: the AdvaPIX TPX3, model B07-W0049 with a 1000 μm CdTe detector [10]. The use of the camera and the selected settings and parameters were part of a previous Master's thesis and were therefore discussed earlier in chapter 3.

4.2 PIXetPro software

The PIXetPro software [21] is used to capture the measured data. There are 2 mayor types of operation. The *frame type* reads out all pixels at the end of exposure. The *pixel type* reads continuously during exposure and only reads out the hit pixels. This project uses the *pixel type* with the *ToT+ToA mode*. That means that for every hit pixel the position, ToT, ToA and FToA are read out continuously [10]. This type and mode, which is also called *event-based*, is chosen because it keeps pixels sensitive at all times [11], which means that multiple events can be detected. The simultaneous reading of ToT and ToA (including FToA) is important because both those values are needed in order to properly reconstruct a Compton interaction.

4.2.1 Registering hit pixels

The output of a measurement using this type and mode is a .t3pa file. Figure 4.3 gives an example of a part of such a file, which returns all the hit pixels in chronological order

with a position parameter and multiple time parameters. The parameters ToA, ToT and FToA were already explained in section 3.3. The other parameters are explained below.

Index	Matrix Index	ToA	ToT	FToA	Overflow
0	318	147924	74	8	0
1	319	147924	135	8	0
2	16	224897	21	9	0
3	12122	713104	3	15	0
4	12121	713102	21	12	0
5	25246	1142774	4	11	0
6	25504	1142774	14	14	0
7	24735	1142774	8	15	0
8	24737	1142774	15	15	0
9	24991	1142774	31	15	0
10	24993	1142774	34	15	0
11	25249	1142774	62	15	0
12	25247	1142774	76	15	0

Figure 4.3: Example of a .t3pa file

Index
This is the index of the hit pixel. This does not give useful information about the event and will therefore be ignored.

Matrix index
The matrix index gives the location of the pixel in the detector. $0 \rightarrow 255$ is the first row. The x-coordinate and y-coordinate can thus be calculated using equation 4.1.

$$\begin{cases} \text{x-coordinate} = \text{remainder of the integer division of } \textit{Matrix Index} \text{ by } 256 \\ \text{y-coordinate} = \text{integer division of } \textit{Matrix Index} \text{ by } 256 \end{cases} \tag{4.1}$$

Overflow
This lets the user know if overflow occurred. In this thesis, the overflow is always zero, so this value can be ignored.

4.2.2 Clustering

In order to reconstruct the Compton interaction, the information of the .t3pa file has to be clustered. This is done by grouping all hit pixels within 200 ns [3] of each other and then dividing this group into clusters based on their location.

All the pixels in a single cluster have to be connected, either horizontally, vertically or diagonally. Figure 4.4 gives a set of examples.

Another possible concern are so-called dead pixels. Those are pixels that are never registered as a hit pixel because their hardware is faulty. Something as simple as a bad connection can cause a dead pixel. Because those pixels do not register, clusters which contain one or more of those pixels are not reliable because they miss a part of the information. To identify dead pixels, the detector was irradiated for 15 minutes with gamma rays of ^{241}Am with an activity of 0.42 MBq. This source was placed on top of the protective cover of the detector right above the sensor. Any pixels that had no counts are considered dead pixels [3]. The PIXetPro software has been given a list of those dead pix-

els. Clusters that (possibly) contain a dead pixel are therefore not listed by the PIXetPro software and do not show up in the .clog file.

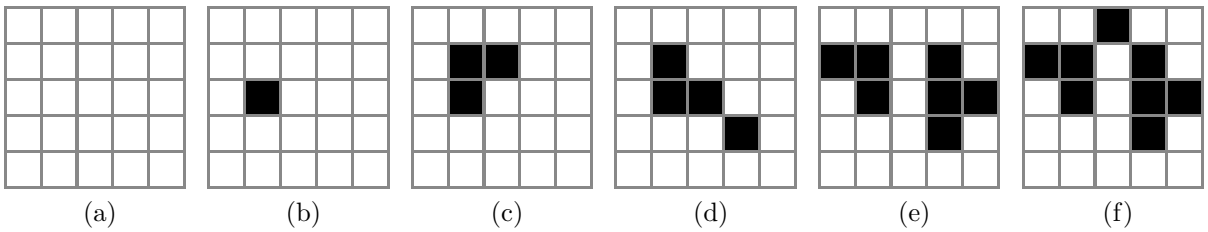


Figure 4.4: Illustration of the clustering principle: a set of examples (a) No hit pixels (b) Single hit (c) Cluster due to horizontal and vertical connection (d) Cluster due to horizontal, vertical and diagonal connection (e) Two separated clusters (f) One cluster due to diagonal connection

Figure 4.5 illustrates clusters that are not listed by the PIXetPro software because they (possibly) contain dead pixels. The dead pixels are red. Figure 4.6 gives two examples where the dead pixel effectuates that it is impossible to know if there is one or two clusters. For that reason those hit pixels are also not listed in the .clog file. Figure 4.7 gives examples where clusters are registered because they are not directly connected to a dead pixel. In 4.7c The upper cluster is listed while the lower cluster is not listed.

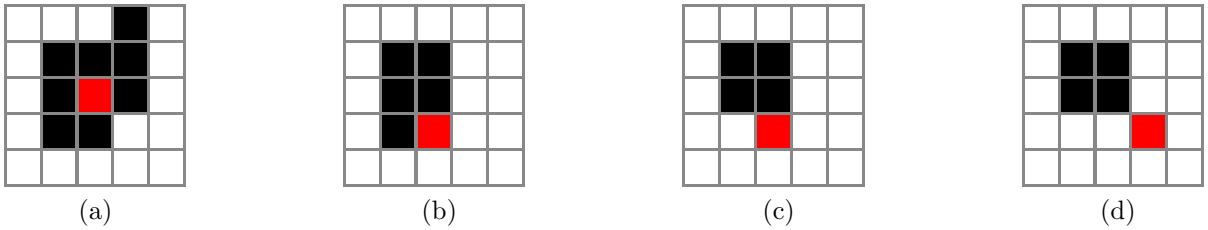


Figure 4.5: Non-registered clusters because of (possible) dead pixels: a set of examples. Some of those (possible) dead pixels are fully enclosed (a), while others might only be connected at the sides (b and c) or at a single corner (d).

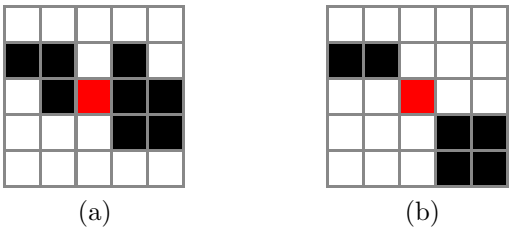
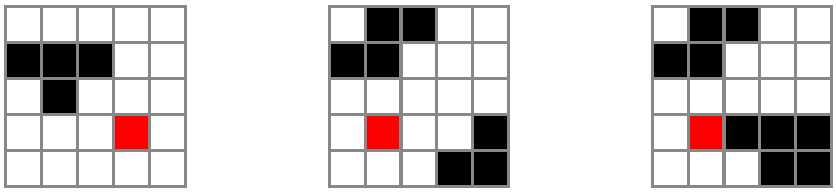


Figure 4.6: Non-registered hit pixels belonging to one or two clusters: a set of examples



(a) (b) (c)

Figure 4.7: Registered clusters: a set of examples

The different steps for clustering are all done by the PIXetPro software. The input for this clustering process is a .t3pa file. The output is a .clog file. A part of such a file is displayed in figure 4.8. Each file consists of different frames. Each frame consist of one or more clusters. Each cluster consists of one or more pixels. A pixel consists of four values: x-coordinate, y-coordinate, energy and extra time. The different elements of a frame are indicated in figure 4.9. Figure 4.10 gives a summary of the conversion between .t3pa files and .clog files. The conversion from ToA and FToA to extra time uses equation 4.2. The arrival time is calculated using equation 3.1.

$$\text{extra time} = \text{arrival time}_{\text{pixel}} - \text{arrival time}_{\text{first pixel}} \tag{4.2}$$

```
Frame 14 (84771978.125000, 0.000000 s)
[200, 36, 5.23944, 0]

Frame 15 (92670239.062500, 0.000000 s)
[105, 219, 24.6191, 0] [104, 219, 26.554, 0] [104, 220, 9.28755, 4.6875] [105, 220, 6.9007, 18.75]

Frame 16 (92895809.375000, 0.000000 s)
[118, 197, 23.4754, 0] [118, 196, 30.7798, 0] [119, 197, 42.5069, 0] [119, 196, 56.9627, 0]

Frame 17 (93415028.125000, 0.000000 s)
[244, 150, 22.0301, 0] [243, 150, 30.8834, 0] [244, 149, 54.0334, 0] [243, 149, 71.8327, 0]
[243, 153, 17.4534, 3.125]

Frame 18 (96654028.125000, 0.000000 s)
[109, 38, 81.1969, 0] [110, 37, 21.9487, 3.125] [110, 38, 7.20992, 10.9375] [111, 37, 5.25665, 15.625]

Frame 19 (99693134.375000, 0.000000 s)
[202, 223, 31.701, 0]
[204, 226, 52.2243, 7.8125]
[214, 217, 21.3909, 18.75] [213, 216, 21.9944, 18.75] [214, 216, 21.1922, 18.75] [213, 217, 18.4721, 20.3125]
[205, 228, 8.82006, 26.5625] [206, 228, 10.543, 26.5625]

Frame 20 (100135301.562500, 0.000000 s)
[73, 93, 24.8573, 0] [71, 93, 93.1639, 0] [72, 93, 93.1639, 0]
```

Figure 4.8: Example of a clog file with 7 illustrative frames that shows that the frames can depict all kinds of different events: single events (frames 14, 15, 16, 18 and 20) as well as multiple events (frames 17 and 19) are possible

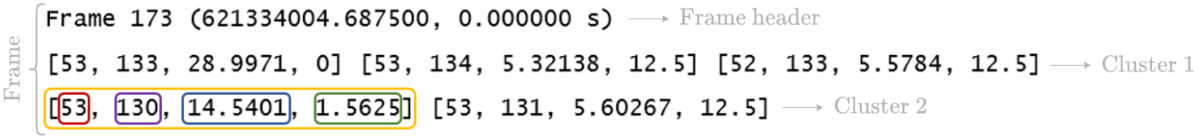
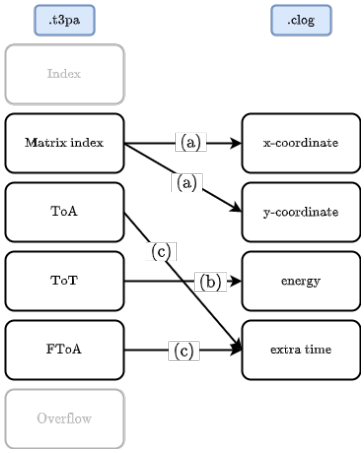


Figure 4.9: Example of a frame consisting of a frame header, cluster 1 and cluster 2: Each frame consists of different pixels and each pixel consists of x-coordinate, y-coordinate, energy and extra time



(a) Equation 4.1
(b) Equation 3.4
(c) Equation 4.2

Figure 4.10: Data conversion from a .t3pa file to a .clog file done by the PiXetPro software [21] prior to the data processing done in this thesis

In a next step, the data from the .clog file is pre-processed so it can be easily used afterwards. Chapter 5 describes this pre-processing.

Chapter 5

Data pre-processing

This chapter describes how the information from a .clog file is processed for further calculations. While the previous chapter 4 described calculations the PIXetPro software [21] did, this chapter describes calculations that were implemented specifically for this thesis using Python [22]. The important scripts and notebooks created for this thesis can be found in the corresponding GitHub repository [23].

5.1 Converting text file to structured data

In order to translate a .clog file that contains a measurement into easily manipulable data, a structured framework needs to be provided. This comes in the form of the class *Measurement*. A .clog file is actually just a text file with a specified structure. This specified structure allows for easy processing of the text file. The class *Measurement* splits the text file into different *Frames*. Each frame consists of one or more *Clusters* and each cluster consists of one or more *Pixels*. Each of those pixels has a x-coordinate, y-coordinate, z-coordinate and energy. Figure 5.1 visualises the data structure.

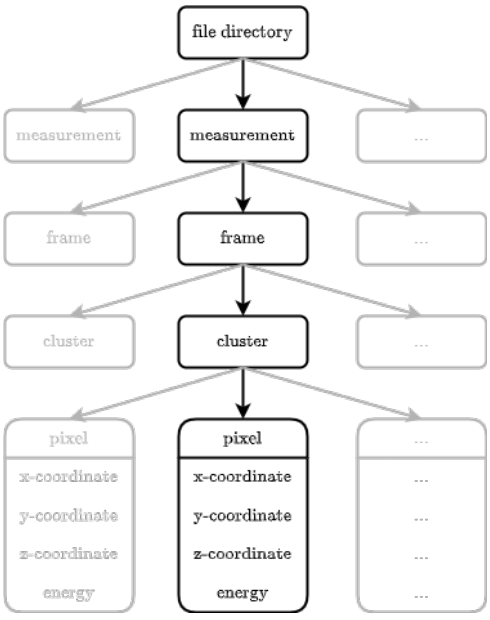


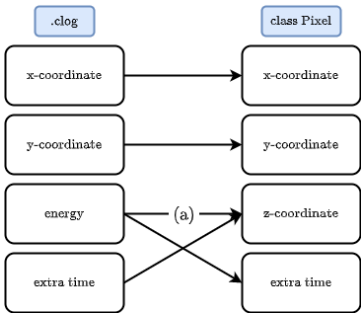
Figure 5.1: Data structure

5.2 Rebuilding structured data

While the splitting from the text file into the different classes goes measurement → frame → cluster → pixel. The rebuilding goes pixel → cluster → frame → measurement.

5.2.1 Pixel

When everything is split, a conversion from the pixel data in .clog form to pixel data in class form is performed. Figure 5.2 visualises this conversion.



(a) Equation 3.6

Figure 5.2: Conversion from .clog file to class Pixel

5.2.2 Cluster

To make a cluster, all the p pixels from this cluster are grouped together. The total energy of the cluster is the sum of the energies of the pixels. The x-coordinate, y-coordinate and z-coordinate are calculated in a weighted manner as shown in equation 5.1. The pixels have a pixel pitch of 55 μm.

$$\begin{cases} E_{cluster} = \sum_{i=1}^{p} E_i \\ x_{cluster} = \frac{55 \cdot \sum_{i=1}^{p} x_i \cdot {E_i}^2}{\sum_{i=1}^{p} {E_i}^2} \\ y_{cluster} = \frac{55 \cdot \sum_{i=1}^{p} y_i \cdot {E_i}^2}{\sum_{i=1}^{p} {E_i}^2} \\ z_{cluster} = \frac{\sum_{i=1}^{p} z_i \cdot {E_i}^2}{\sum_{i=1}^{p} {E_i}^2} \end{cases} \tag{5.1}$$

5.2.3 Frame

A measurement consists of a lot of frames. The total number of frames is referred to as f. Most of those frames do not represent a Compton interaction. The goal is to select these frames that do represent a Compton interaction. The rules for selecting those frames are rather simple at first sight.

1. The frame must contain exactly two clusters.

2. The total energy of the frame must lie between E_{min} and E_{max}.

In an ideal situation the total energy of the frame would be E_γ. In practice, the total energy will deviate from the ideal value of E_γ. The parameter R_e, also referred to as energy range, determines how much the energy of the frame may deviate from the ideal energy E_γ: E_{min} [kev] = E_γ [kev] − R_e [kev] and E_{max} [kev] = E_γ [kev] + R_e [kev]. In chapter 9 a good value for R_e will be determined in the case of ^{137}Cs where E_γ = 662 keV.

Before selection based on these two criteria can be executed, correction for X-ray fluorescence (XRF) must be applied.

Correcting for XRF

XRF is the phenomenon where a γ ray excites an atom which leads to an XRF photon being emitted. In this thesis, it is about internal fluorescence of the CdTe sensor which means the γ ray excites a Cd atom or a Te atom. This emitted XRF photon is then detected in an other pixel which can lead to a new cluster being formed with an energy E_{XRF}. This also means that the energy of the original cluster $E_{original\,cluster}$ is decreased with E_{XRF}. Therefore a correction for XRF has to be applied. This is implemented as follows: If a frame with exactly 3 clusters has exactly one cluster with an energy below 40 keV then its energy is added to the the one cluster of the remaining two clusters that is located closest to the XRF cluster. The XRF cluster is then removed from the list. This results in a frame with two clusters.

In a frame that originally has two clusters, a cluster with an energy below 40 keV will also be interpreted as an XRF cluster, which means that cluster does not represent a Compton interaction. This leads to the definition of usable frames.

Usable frames

Every time a frame is constructed, a list of clusters is made and the total energy of those clusters is calculated. Then the frame will be corrected if necessary. The last step is checking if the (corrected) frame is usable. This means that the frame

1. has exactly two clusters,
2. has a total energy between E_{min} and E_{max},
3. has no cluster with an energy below 40 keV.

The number of usable frames is referred to as uf.

5.2.4 Measurement

When 'rebuilding' a measurement, the code will build every frame and add it to a list that contains all the frames. In that same iteration, it will check if the frame is usable and if so, will add it to the list of usable frames. It also has a function — that does not run on initialisation — that calculates the axes and angles of the Compton cones for every usable frame. This function will be discussed in section 5.3 .

5.3 Compton processing

For every usable frame, the possible cones will be calculated. For this, the normalised scatter axis $\vec{u}$ and opening angle θ are necessary. The opening angle is calculated using the Compton equation (2.2). The normalised scatter axis $\vec{u}$ is calculated using equation 5.2. $[x_0, y_0, z_0]$ are the coordinates of the first interaction, while $[x_1, y_1, z_1]$ are the coordinates of the second interaction.

$$\vec{u} = \frac{[x_0 - x_1, y_0 - y_1, z_0 - z_1]}{\sqrt{(x_0 - x_1)^2 + (y_0 - y_1)^2 + (z_0 - z_1)^2}} \tag{5.2}$$

Relation between twin cones

Some usable frames produce only one cone, while others produce two. When two cones are created, these cones are called twin cones. This section takes a little detour to explore the relation between twin cones. The equation that results in the opening angle of the cones is equation (5.3), which is the same equation as 2.2 [4].

$$\cos(\theta) = 1 - m_e c^2 \frac{E_e}{E_\gamma(E_\gamma - E_e)} \tag{5.3}$$

A cosine can only have a value between -1 and 1. To determine the associated energies E_e that result in the cosine being -1 and 1, equation 5.3 is filled in and simplified. The results are shown below.

$$-1 = 1 - m_e c^2 \frac{E_e}{E_\gamma(E_\gamma - E_e)} \Rightarrow E_e = \frac{2 \cdot E_\gamma^2}{m_e c^2 + 2 \cdot E_\gamma}$$

$$1 = 1 - m_e c^2 \frac{E_e}{E_\gamma(E_\gamma - E_e)} \Rightarrow E_e = 0$$

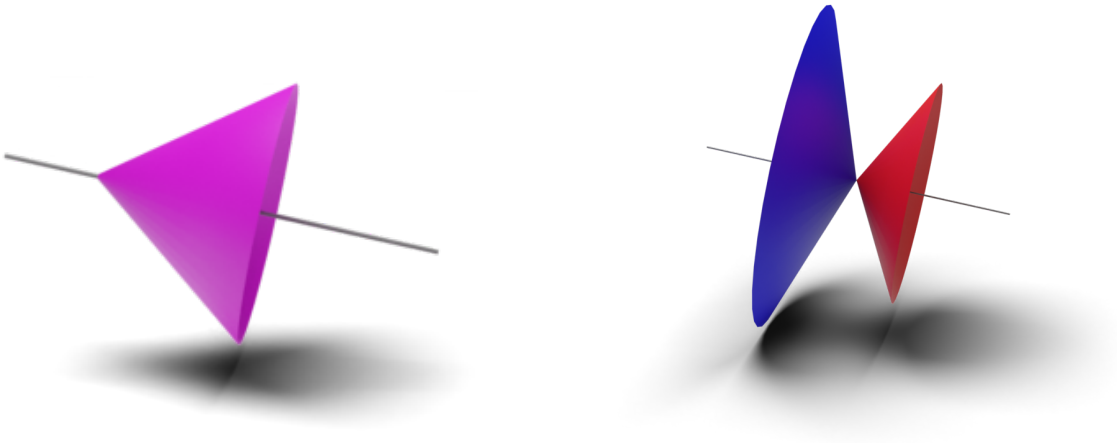
To find the maximum energy for ^{137}Cs, the symbols are substituted by their known values: $E_\gamma = 662$ keV, $m_e c^2 = 511$ keV.

$$E_e = \frac{2 \cdot E_\gamma^2}{m_e c^2 + 2 \cdot E_\gamma} \Rightarrow E_e = \frac{2 \cdot 662^2}{511 + 2 \cdot 662} \text{ keV} = 477.65 \text{ keV}$$

While the minimum value for E_e is 0 keV according to these calculations, it was established in 5.2.3 that the minimum value for a usable frames is 40 keV. This means that only frames with energies between 40 keV and 477.65 keV result in a cone. When only one of the two energies lie in that range, there is only one cone. When both energies lie in that range, twin cones are formed.

Cones constructed from an energy of (662 - 477.65) keV = 184.35 keV have an opening angle of $0.792 = 45.4°$. This means that cones with an opening angle of 45.4° or lower are single cone while cones with a greater opening angle have a twin cone.

Figure 5.3a shows a single cone with an opening angle of 56.3°. Figure 5.3b illustrates twin cones with an opening angle of 79.3° (blue) and 74.5° (red).



(a) Single cone with an opening angle of 36.9°

(b) Twin Cones with opening angles of 79.3° (blue) and 74.5° (red)

Figure 5.3: Difference in opening angle between a single cone and twin cones

Chapter 6

Orientation and convention

This chapter describes the different orientations and conventions used in this thesis. The main goal of this chapter is to be a guide to understand the rest of the thesis and the implementation of the code.

6.1 Cartesian coordinate system

The orientation of the x, y and z-axis according to the convention of the detector is illustrated in figure 6.1.

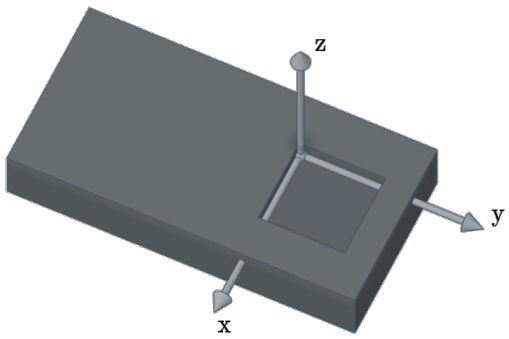


Figure 6.1: Orientation of x, y and z-axis according to detector convention

Figure 6.2 illustrates the new convention that is used in this thesis. This convention makes sure that the point [1,0,0] is located right in front of the detector when $\phi = 0$ and $\tau = 0$. The created coordinate system is a left-handed coordinate system. [1,0,0] is the unit vector that describes the x-axis, [0,1,0] is the unit vector that describes the y-axis and [0,0,1] is the unit vector that describes the z-axis.

When drawing the cones, the apex is always located at the middle of the sensor and not at the location of the cluster that represents the first interaction. This introduces a small error but makes computation significantly easier. The coordinate system is thus located at the centre of the sensor as illustrated in figure 6.3.

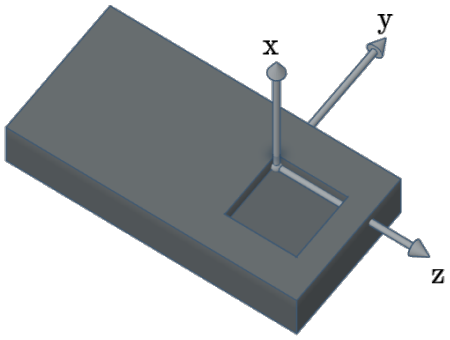


Figure 6.2: Orientation of x, y and z-axis according to the thesis convention

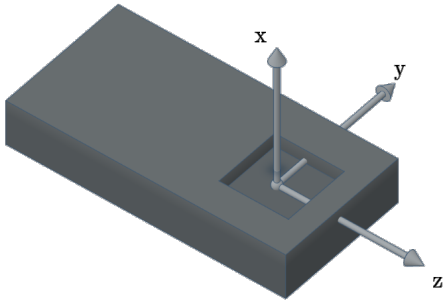


Figure 6.3: Orientation of the axes in the centre according to the thesis convention

6.2 Axes of rotation and projection sphere

Figure 6.4 displays the ϕ-axis and τ-axis. The Compton camera will revolve around these axes, following the right-hand rule, to carry out the measurements.

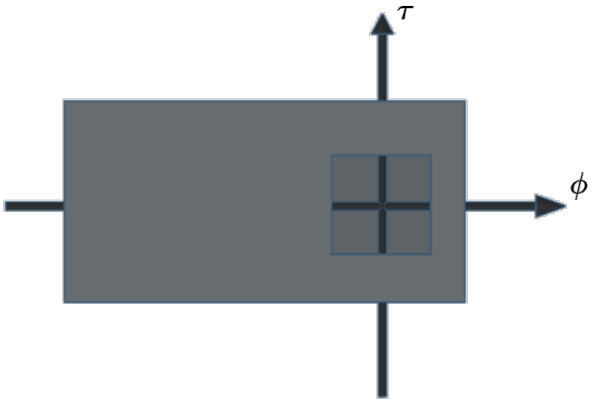


Figure 6.4: Orientation of ϕaxis and τ axis

The photo of the measurement setup (figure 6.5) gives a better understanding of how these axes rotate. The photo illustrates that when there is a rotation about the τ-axis, the ϕ-axis also rotates. However, when there is a rotation around the ϕ-axis, the τ-axis does not rotate along.

Figure 6.5: Photo of the measurement setup

ϕ and τ are also used to create the projection sphere. In order to create a sphere ϕ ranges from -π to π and τ ranges from -$\frac{\pi}{2}$ to $\frac{\pi}{2}$. In order to draw the sphere in Python and in order to calculate the intersections between the sphere and the cones, a conversion from spherical coordinates to Cartesian coordinates has to be made. This is done using equation 6.1.

$$\begin{cases} x = r \cdot \sin(\tau + \frac{\pi}{2}) \cdot \cos(\phi) \\ y = r \cdot \sin(\tau + \frac{\pi}{2}) \cdot \sin(\phi) \\ z = r \cdot \cos(\tau + \frac{\pi}{2}) \end{cases} \quad \text{with} \quad \begin{cases} \tau \text{ in degrees} \\ \phi \text{ in degrees} \\ \text{r} = 1 \end{cases} \tag{6.1}$$

Figures 6.6, 6.7 and 6.8 illustrate the angular difference of each axis with the projection sphere.

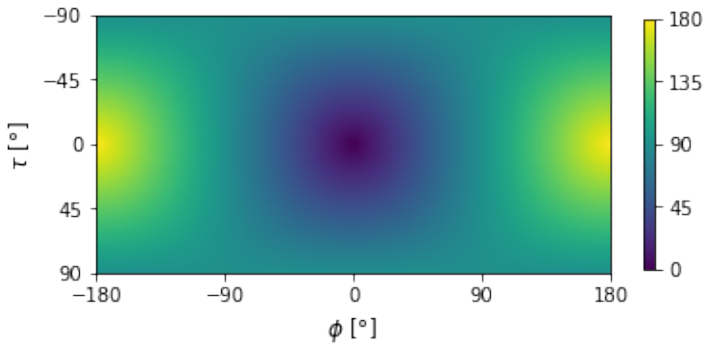


Figure 6.6: Angular difference of $\vec{u}$ = [1,0,0] with the projection sphere

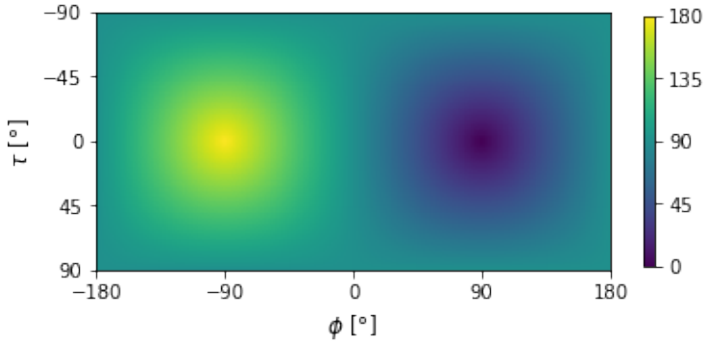


Figure 6.7: Angular difference of $\vec{u}$ = [0,1,0] with the projection sphere

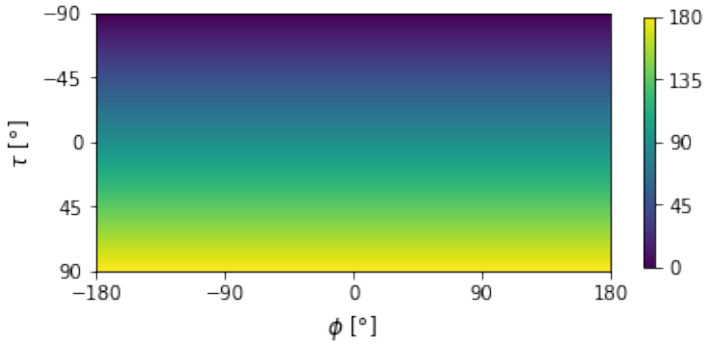


Figure 6.8: Angular difference of $\vec{u}$ = [0,0,1] with the projection sphere

6.3 Alternative convention

An alternative convention could be to follow the model of a globe. This would mean that θ ranges from -π to π and that ϕ ranges from -$\frac{\pi}{2}$ to $\frac{\pi}{2}$.

The new convention for the Cartesian coordinate system would then be constructed in such a way that the centre of the earth is the origin of the coordinate system, the x-axis 'pierces' Africa and that the North Pole and South Pole are located on the z-axis. The position of the y-axis is then chosen so a right-handed coordinate system is created. Figure 6.9 and figure 6.10 illustrate this.

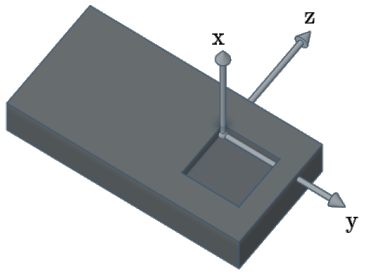


Figure 6.9: Alternative convention for the Cartesian coordinate system

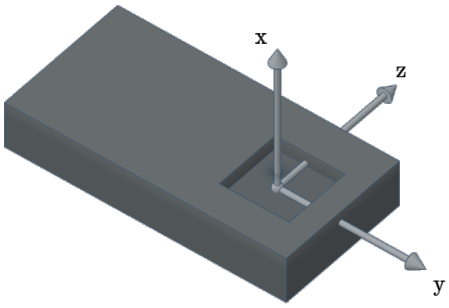


Figure 6.10: Alternative orientation of the coordinate system with the axes in the centre

Chapter 7

Taking a look at the data

This chapter takes a closer look at the gathered data. The number of measurements is referred to as n. There are 276 measurements in the used data set. Each of those has a measurement time of 1 hour. Appendix A gives an overview of all the measurements: ϕ ranges from -55° to 0° and τ ranges from -55° to 55°. The 'correct' intersection point according to the measurements will be referred to as $P_m = [\phi_m, \tau_m]$. The calculated point of intersection will be referred to as $P_c = [\phi_c, \tau_c]$.

Figure 7.1 visualises the number of frames for every measurement. The circular pattern indicates that the number of frames is higher when P_m is closer to [0°,0°]. Figure 7.2 visualises the expected distribution. This figure was made using the absorption probability in function of the incident angle. The code to create this figure used the xraylib library [24]. The probability for every incident angle was then multiplied by the sum of the number of frames for every incident angle divided by sum of the probability for every incident angle. Comparing these two figures leads to the conclusion that the circular pattern is indeed expected, but that the number of frames has a steeper decline than expected.

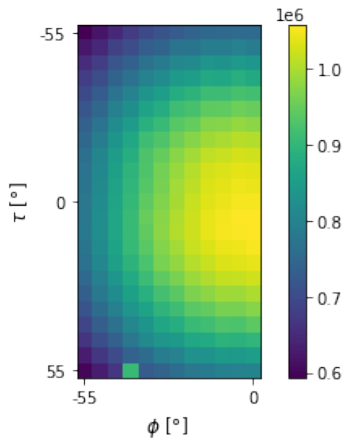


Figure 7.1: Number of frames for every measurement

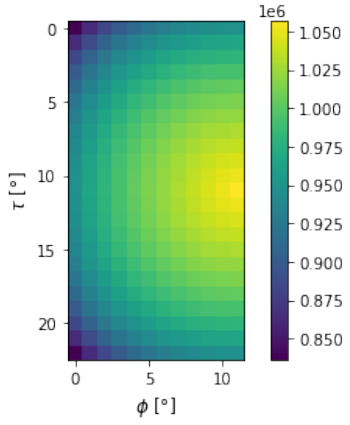


Figure 7.2: Expected distribution of the number of frames based on absorption probability in function of the incident angle

The measurement with $P_m = [-40°, -55°]$ does not fit in the pattern. There can be several causes. One possibility is a equipment malfunction, but an other — probably

more likely — possibility is a change in environment. The measurement setup was set up in a place that is accessible to other people and where other experiments are done. It can be that an other person was passing by with another source, that a laser hit the detector, or any other change in environment that happens when multiple people use the same lab. Because this measurement can be labelled as an outlier, it could be excluded from the measurements. Because it was only 1 of 276 measurements and the results of that measurement were in line with the other measurements, the measurement was **not** excluded from the data.

Figure 7.3 visualises the number of usable frames for every measurement for $R_e = 30$ keV. Two things can be observed: first, there is no circular pattern as there was in figure 7.1 which means there is no directly proportional relationship between the number of frames and the number of usable frames and second, there are more usable frames for theta in range [30°,35°]. This figure does not have a clear pattern as was the case for figure 7.1. That is because only one hit is not enough, but a complete Compton interaction has to be recorded. The mathematics that predict the pattern of which angles will have the most usable frames are very complex. Therefore, further analysis of this figure is not a part of the scope of this thesis.

Figure 7.4 visualises the number of cones that intersect in P_c for $R_e = 30$ keV and $\sigma = 2°$ for simple addition for every measurement. Although this figure is not exactly the same as figure 7.3, the same patterns can be found. This is logical as more usable frames lead to more cones which in turn leads to more cones that can and will intersect.

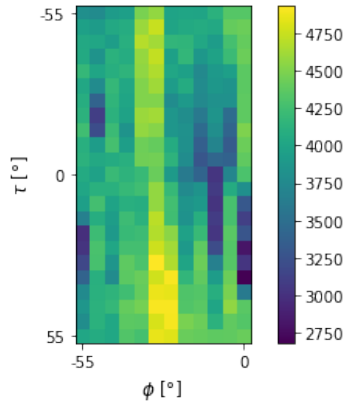


Figure 7.3: Number of usable frames for every measurement ($R_e = 30$ keV)

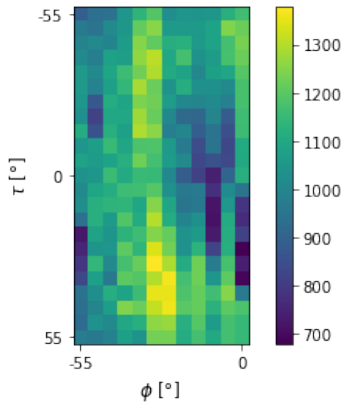


Figure 7.4: Number of cones intersecting in P_c for every measurement for simple addition with $R_e = 30$ keV and $\sigma = 2°$

Comparing figure 7.1 and figure 7.3 illustrates that only a small number of frames are considered usable frames. The percentage of frames considered usable frames is referred to as $uf\%$ The average percentage for all n measurements of frames used is calculated using equation 7.1. This results in 0.479% for $R_e = 30$ keV and 0.572% for $R_e = 35$ keV.

$$uf\%_n = \frac{1}{n} \sum_{i=1}^{n} \frac{uf_n}{f_n} \tag{7.1}$$

The next step is analysing the number of cones that intersect in P_c, referred to as *ic*.

It is important to remember that a cone not just 'intersects or not intersects'. Every cone has a value between 0 and 1 for for P_c and not every usable frame creates a cone that intersects in P_c. Taking every measurement into account, the average percentage of frames that result in a cone that intersect in P_c, referred to as $ic\%$, can be calculated using equation 7.2.

$$ic\%_n = \frac{1}{n} \sum_{i=1}^{n} \frac{ic_n}{f_n} \tag{7.2}$$

That results in $ic\%_n = 0.126\%$ for simple addition with $R_e = 30$ keV and $\sigma = 2°$. That also means that for simple addition with $R_e = 30$ keV and $\sigma = 2°$, 0.126/0.479 = 26,3% of the usable frames result in a cone that intersects in P_c.

Chapter 8

Back-projection

While section 2.3 visualises back-projection using a projection plane, in this thesis a projection sphere is used. Because the source is always the same distance from the detector, it is logical to have a projection surface that is also always the same distance from the detector. The distance from the detector to the source is 30 cm. This would mean that a sphere with a radius of 30 cm is needed. However, because the distance is already known, the position is fully defined using the angles ϕ and τ. Therefore the radius of the sphere is irrelevant. To simplify further calculations, a unit sphere will be used.

Figure 8.1 illustrates the concept. Each cone is reconstructed based on the data from one Compton event (see section 2.3). To keep the figure clear, only single cones are used. Every cone intersects with the projection sphere, resulting in a circle. All these circles — and thus also the cones — intersect in one point on the sphere. The angles ϕ_c and τ_c that correspond with that point are the points that define P_c. In this example, all the cones neatly intersect in one point. This will not be the case with real data. Because it is not practical to use a 3D sphere for calculations, the sphere will be displayed as a 2D map, very similar as to how a globe unfolds into a world map. This also means that while the intersection of a cone with the sphere is a circle in 3D space, it will not always look like a sphere on a 2D map.

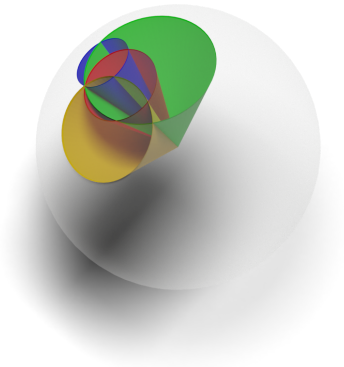


Figure 8.1: Back-projection using a projection sphere

8.1 Preparing for back-projection

8.1.1 Creating the projection sphere

The unit sphere will be parameterised using the angles ϕ and τ. It is discretised with a resolution a_S [°]. The sphere is defined using equation 8.1. The reasoning behind this equation is explained in chapter 6. To create a full sphere ϕ needs to range from $-\pi$ to π and τ from $\frac{-\pi}{2}$ to $\frac{\pi}{2}$. a_S defines how many values there are for ϕ and τ using equation 8.2. A unit sphere with $a_S = 1°$ therefore has 361 values for ϕ and 181 values for τ. For every combination of ϕ and τ there is a x, y and z-value. That means the sphere is defined as a 3D matrix of 3 rows, 181 columns and 361 layers. In order to later use this sphere in matrix multiplication, the sphere has to be unravelled into a 2D matrix of 3 rows (x, y and z) and 181 x 361 = 65341 columns. This 2D matrix will be referred to as S. For the purpose of illustration, a unit sphere with $a_S = 60°$ is assumed. This leads to a 3D matrix of 3 rows, 4 columns and 7 layers as illustrated in figure 8.2. Figure 8.3 illustrates the unravelled version with 3 rows and 28 columns.

$$\begin{cases} x_{sphere} = \sin(\tau) + \frac{\pi}{2} \cdot \cos(\phi) \\ y_{sphere} = \sin(\tau) + \frac{\pi}{2} \cdot \sin(\phi) \\ z_{sphere} = \cos(\tau + \frac{\pi}{2}) \end{cases} \tag{8.1}$$

$$\begin{cases} \text{number of } \phi \text{ values} = \frac{360}{a_S} + 1 \\ \text{number of } \tau \text{ values} = \frac{180}{a_S} + 1 \end{cases} \tag{8.2}$$

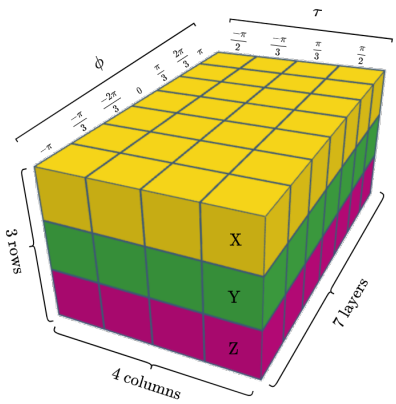


Figure 8.2: The values of the discretised projection sphere as a 3D matrix

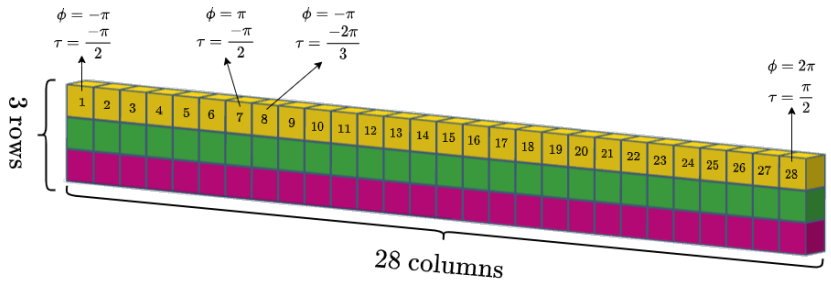


Figure 8.3: The values of the unravelled sphere S as a 2D matrix

8.2 Creating a single back-projection

8.2.1 Calculating the angular difference

The angle between a point on the sphere and the cone axis $\vec{u}$ is defined as δ_{axis}. This value can be calculated for every single point on the sphere — meaning for every combination of ϕ and τ — using equation 8.3. This can be visualised by creating a map where ϕ varies horizontally and τ vertically. Figure 8.4 illustrates this with $\vec{u} = [1,0,0]$. This can be interpreted as a cone with $\vec{u} = [1,0,0]$ and $\theta = 0°$.

$$\delta_{axis}\ [°] = \arccos(\vec{u} \cdot S) \cdot \frac{180°}{\pi} \tag{8.3}$$

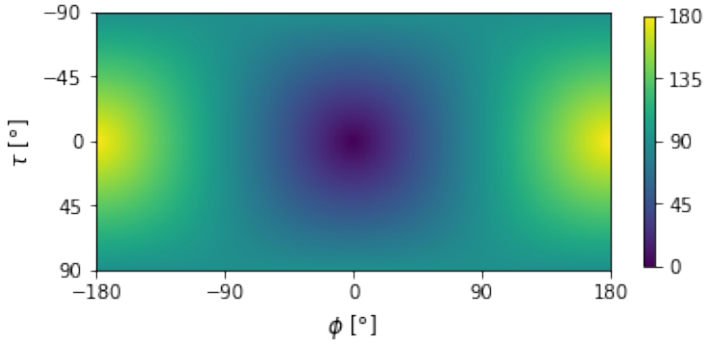


Figure 8.4: 2D map of the angular difference for a cone with $\vec{u} = [1,0,0]$ and $\theta = 0°$

To construct a 2D map for a cone with $\theta > 0°$, the difference of δ_{axis} and θ, defined as δ_{total}, has to be calculated using equation 8.4. This results in figure 8.5.

$$\delta_{total}\ [°] = \delta_{axis}\ [°] \text{ - } \theta\ [°] \tag{8.4}$$

Although figure 8.5 is correct, it is not very easy to interpret. To improve visualisation, the colours need to be shifted so that dark blue represent 0 again. This is achieved by taking the absolute value of δ_{total}, referred to as $\delta_{total,abs}$, using equation 8.5. Figure 8.6 visualises the result. This figure with the dark blue circle — which is the projection of the cone — is significantly easier to interpret than figure 8.5.

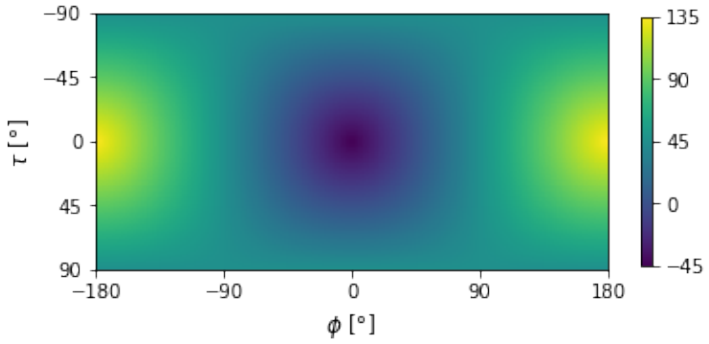


Figure 8.5: 2D map of the angular difference for a cone with $\vec{u}$ = [1,0,0] and $\theta = 45°$

$$\delta_{total,abs}\ [°] = |\delta_{axis}\ [°] - \theta\ [°]| \tag{8.5}$$

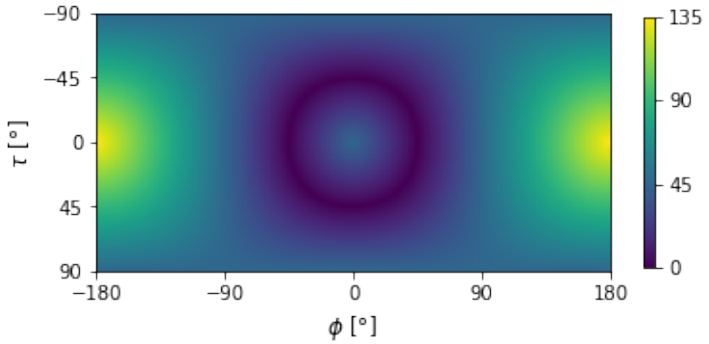


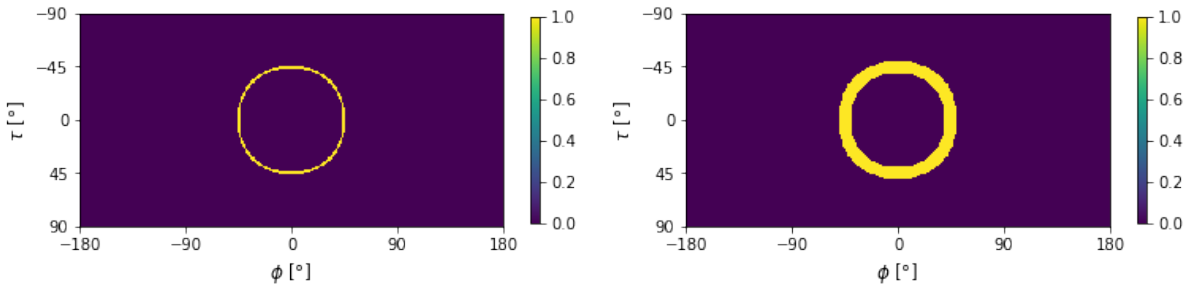
Figure 8.6: 2D map of the absolute angular difference for a cone with $\vec{u}$ = [1,0,0] and θ = 45°

8.2.2 Introducing a threshold

To highlight the shape of the projection of the cone, a threshold t can be introduced so that everything beneath that threshold is marked as 1 and everything above that threshold as 0. This leads to equation 8.6. The result of this equation is referred to as δ_t. In a first example, $t = 1°$. That means that every point with $\delta_{total,abs} > 1°$ is set to 0 and every point with $\delta_{total,abs} \leq 1°$ is set to 1. This results in figure 8.7a. A small t results in a thin circle. This results in the most precise projection, but difficulties might arise later on when the intersection between projections has to be found, because the margin for error is small. A bigger t solves this problem. This results in figure 8.7b where $t = 5°$.

$$\begin{cases} \delta_t = 0 & \text{if } \delta_{total,abs} > t \\ \delta_t = 1 & \text{if } \delta_{total,abs} \leq t \end{cases} \tag{8.6}$$

While this approach succeeds in making the circle thicker and thus having a bigger margin of error, it treats all the points of the 'circle' as 'equally good' while the points that 'survive' a lower t are actually 'better' than the ones only surviving a bigger t. This problem can be solved by not simply assigning 1 to the values smaller than t, but by



(a) (b)

Figure 8.7: 2D map of the threshold function of the angular difference as defined in equation 8.6 for a cone with $\vec{u}$ = [1,0,0], $\theta = 45°$ and (a) $t = 1°$ (b) $t = 5°$

assigning values according to a linear gradient using equation 8.7. The result is referred to as $\delta_{t,linear}$.

$$\begin{cases} \delta_{t,linear} = 0 & \text{if } \delta_{total,abs} > t \\ \delta_{t,linear} = \frac{t \text{ - value}}{t} & \text{if } \delta_{total,abs} \leq t \end{cases} \tag{8.7}$$

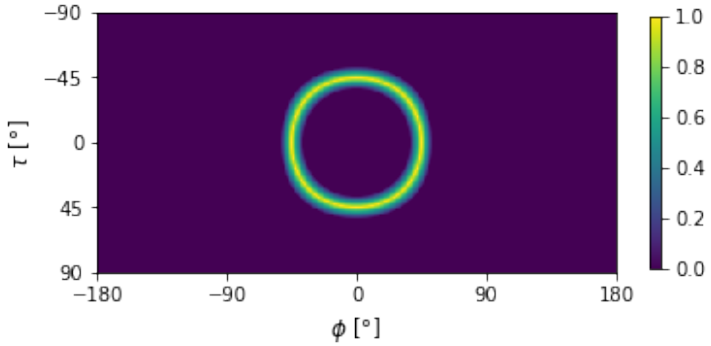


Figure 8.8: 2D map of the linear threshold function of the angular difference as defined in equation 8.7 for a cone with $\vec{u}$ = [1,0,0], $\theta = 45°$ and $t = 8°$

8.2.3 Introducing the Gaussian equation

The Gaussian equation

A Gaussian equation is an equation of the form:

$$f(x) = a \cdot e^{\left(-\frac{(x-b)^2}{2 \cdot c^2}\right)} \qquad \text{with real constants a, b and non-zero c.} \tag{8.8}$$

Figure 8.9 visualises the graph of a Gaussian equation. It is shaped like a symmetric bell curve. The parameters a, b and c define the shape of that bell curve. Parameter a defines the height of the peak, parameter b defines the position of the centre of the peak and parameter c defines the width of the peak. Parameter c is also known as the standard deviation and will from now on be represented by the letter σ.

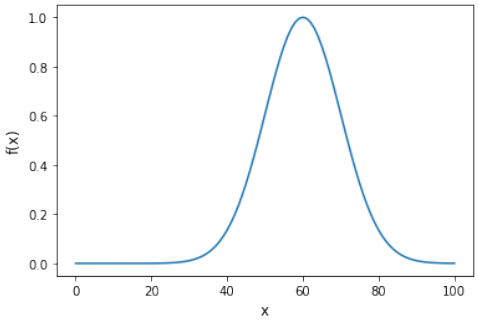


Figure 8.9: Graph of a Gaussian function with a = 1, b = 60 and c = σ = 10

Applying the Gaussian equation

Instead of the linear threshold equation defined in 8.7, a Gaussian equation can be applied to highlight the projection of the cone. Equation 8.9 calculates the Gaussian difference δ_g. To do this, the general parameters a, b, c and and the variable x have to be replaced by their specific counterparts. This means that x = δ_{axis} and b = θ. These parameters are fixed for a certain cone. The choice for parameter a is not very relevant because the comparison will always be relative. Therefore, a = 1. The parameter c, which is already defined as the standard deviation σ, can be varied in order to get a smaller or wider peak which results in a thinner or thicker back-projection that in turn results in a smaller or larger margin of error when calculating the intersection of cone projections. Figure 8.10 displays the 2D map of the Gaussian difference of a cone with $\vec{u}$ = [1,0,0], $\theta = 45°$ and σ = 3. This figure is similar to figure 8.8. To give better insight into the differences, figure 8.11 compares the descent of the two.

$$\delta_g = e^{\frac{-|\delta_{axis} - \theta|^2}{2 \cdot \sigma^2}} \tag{8.9}$$

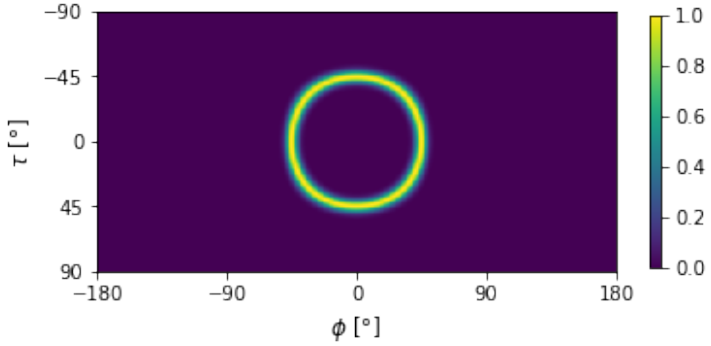
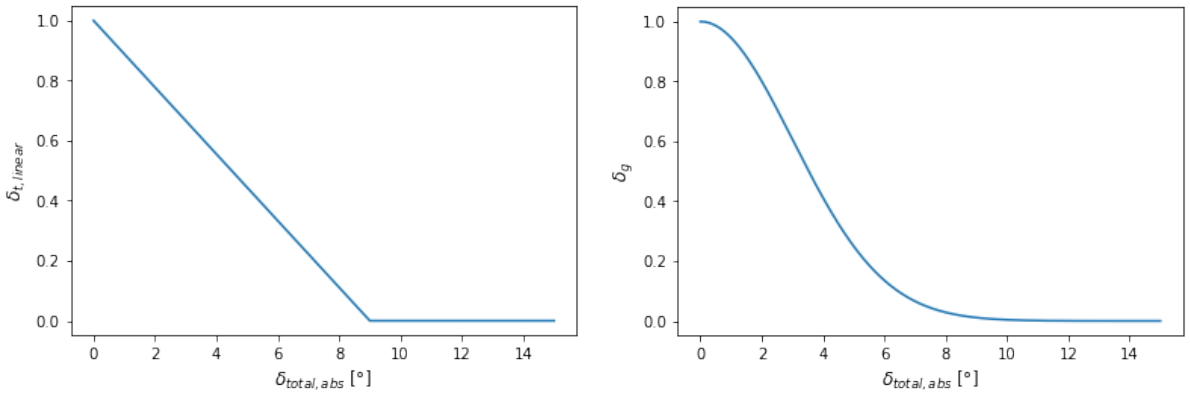


Figure 8.10: 2D map of the Gaussian difference for a cone with $\vec{u}$ = [1,0,0], $\theta = 45°$ and $\sigma = 3°$

Mathematically, δ_g always lies within the range]0,1]. However, when calculating δ_g in Python, the range is [0,1]. A point that is located exactly on the back-projection of the cone will result in $e^{\frac{-|0|^2}{2 \cdot 3^2}} = e^0 = 1$. A point that is located far from the back-projection, for example 135°, will result in $e^{\frac{-|135|^2}{2 \cdot 3^2}} = e^{-1800} = 1.86 \cdot 10^{-782}$. Because Python cannot handle numbers that small, it rounds $1.86 \cdot 10^{-782}$ to 0, even though e^x can mathematically never be 0. In this explanation, $\sigma = 3°$ was assumed.

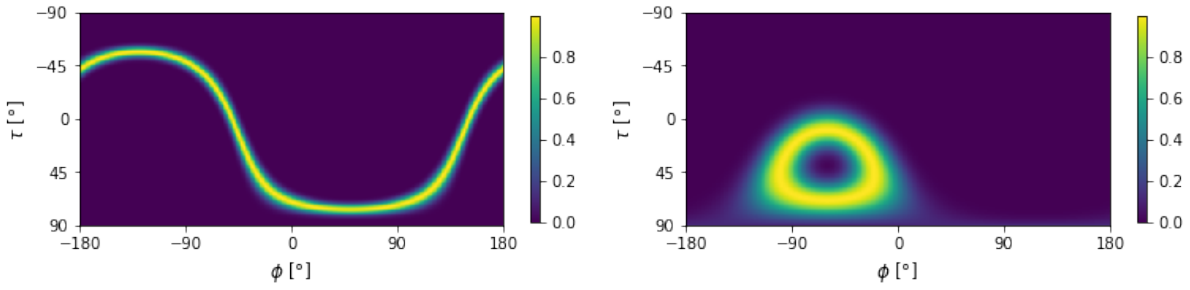


(a) (b)

Figure 8.11: Difference in descent for (a) $\delta_{t,linear}$ with $t = 9°$ and (b) δ_g with $\sigma = 3°$

8.2.4 Example for $\vec{u} \neq [1, 0, 0]$

The previous examples all used $\vec{u}$ = [1,0,0] and $\theta = 45°$ because this results in a clear 'round' figure that is easy for explanation. Because not all back-projections have the same shape, figure 8.12a illustrates the Gaussian difference for a cone with $\vec{u}$ = [0.5857, 0.7095, 0.3919], $\theta = 100°$ and $\sigma = 3°$ and figure 8.12b for a cone with $\vec{u}$ = [0.3873, -0.6708, -0.6325], $\theta = 30°$ and $\sigma = 10°$. This illustrates that back-projections can have very different shapes than the one used in all the previous examples, while also illustrating the impact of σ: figure 8.12b with $\sigma = 10°$ displays a significantly thicker back-projection than figure 8.12a with $\sigma = 3°$.



(a) (b)

Figure 8.12: 2D map of the Gaussian difference for a cone with (a) $\vec{u}$ = [0.5857, 0.7095, 0.3919], $\theta = 100°$ and $\sigma = 3°$ and (b) $\vec{u}$ = [0.3873, -0.6708, -0.6325], $\theta = 30°$ and $\sigma = 10°$

This section illustrated different methods of creating a 2D map, starting from calculating just the angular difference, through threshold equations all the way up to the Gaussian implementation. From this point on, all the calculations will be done with the 2D maps created with the Gaussian implementation, unless mentioned otherwise.

8.3 Sum of back-projections

In order to determine the angles ϕ and τ that define the position of the source, the point where the different back-projections intersect has to be calculated. This is done

by adding up the different 2D maps and locating the point with the highest value. The angular difference between P_m and P_c will be referred to as ε_a [°] and is calculated using equation 8.12, which uses equation 8.10 and equation 8.11. In the equations below, all the angles are in rad because that is easier for the calculations, but when discussing ε_a, degrees is used. Degrees can be easily transformed in rad and the other way around because 1 rad = $\frac{180}{\pi}$ °.

$$\begin{cases} x_m = \sin\left(\tau_m + \frac{\pi}{2}\right) \cdot \cos\left(\phi_m\right) \\ y_m = \sin\left(\tau_m + \frac{\pi}{2}\right) \cdot \sin\left(\phi_m\right) \\ z_m = \cos\left(\tau_m + \frac{\pi}{2}\right) \end{cases} \quad \text{with} \begin{cases} \tau_m \text{ in rad} \\ \phi_m \text{ in rad} \end{cases} \tag{8.10}$$

$$\begin{cases} x_c = \sin\left(\tau_c + \frac{\pi}{2}\right) \cdot \cos\left(\phi_c\right) \\ y_c = \sin\left(\tau_c + \frac{\pi}{2}\right) \cdot \sin\left(\phi_c\right) \\ z_c = \cos\left(\tau_c + \frac{\pi}{2}\right) \end{cases} \quad \text{with} \begin{cases} \tau_c \text{ in rad} \\ \phi_c \text{ in rad} \end{cases} \tag{8.11}$$

$$\varepsilon_a = [x_c, y_c, z_c] \cdot [x_m, y_m, z_m] \qquad \text{with } \varepsilon_a \text{ in rad} \tag{8.12}$$

8.3.1 Simple addition

A first option is to interpret twin cones simply as two different cones and making a 2D map for each one of the cones. Then all these maps are added up. This is what is called 'simple addition'.

8.3.2 Twin addition

A different option is to keep twin cones together and to make 1 2D map for every pair of twin cones — a single cone still has its own map. This is called 'twin addition'. The map for twin cones is **not** made by simply adding the 2D maps for every cone as this would give the same result as simple addition. Instead, the maximum is taken for every point. This is done because it prevents points of the wrong cone being taken into account where possible. Because the value for one of the cones is 0 for most of the points, this will not make a significant difference for every point. Only when the two back-projections are close to each other this will make a difference. The results in chapter 9 illustrate that this new method does indeed give better results than simple addition.

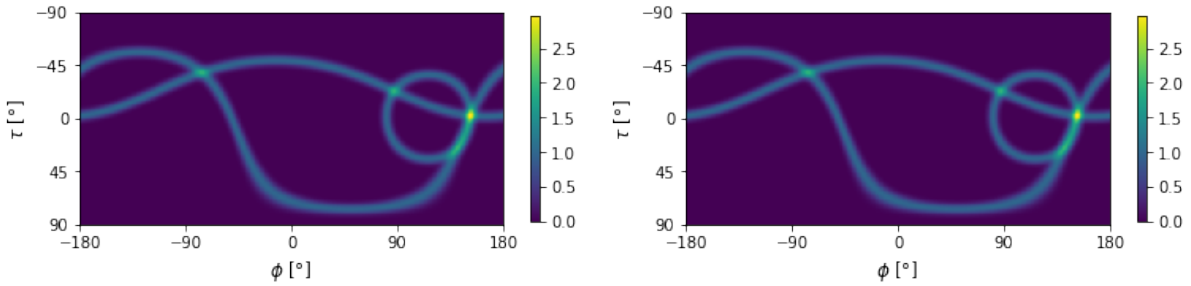
8.3.3 Example with artificial data

Table 8.1 gives an overview of artificial cones defined by their normalised scatter axis $\vec{u}$ and opening angle θ. The correct value for this set of cones is $P = [153°, -3°]$. The table consists of 3 pairs of twin cones. Cone a is the correct cone, while cone b is the incorrect twin cone. Figure 8.13 illustrates the sum of the back-projections of the correct cones, $P_c = [153°, -3°]$. Figure 8.14 illustrates the sum of the back-projections of the incorrect twin cones, $P_c = [164°, -12°]$. Figure 8.15 illustrates the sum of all the back-projections, correct and incorrect ones, $P_c = [153°, -3°]$. The three figures illustrate two things. First, simple addition and twin addition give the same results in this case — on a small amount

of artificial data. Second, the sum of back-projections of correct and incorrect cones give different results, but when both are combined, the result is the same as the one with only the correct cones. This is because the correct cones intersect neatly in 1 point, while the incorrect twin cones did not. This is corroborated by the values of the colour scale.

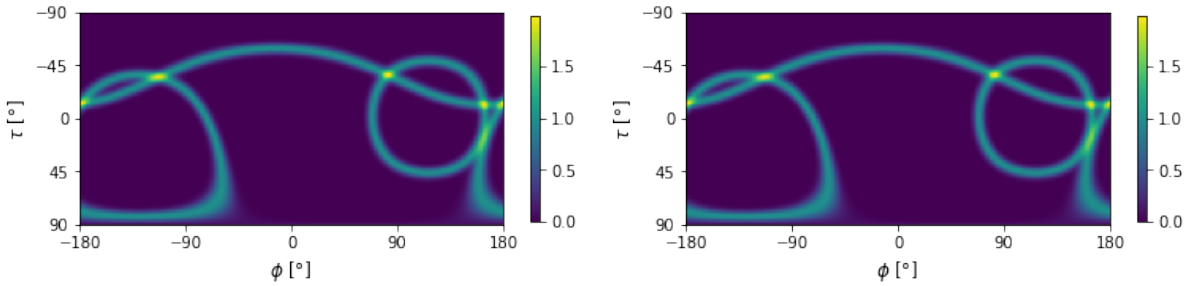
Table 8.1: Normalised scatter axes $\vec{u}$ and opening angles θ of the selected cones

Cone	$\vec{u}$	θ [°]
1a	[0.5857, 0.7095, 0.3919]	100.2363
1b	[-0.5857, -0.7095, -0.3919]	60.3912
2a	[0.4525, -0.8914, -0.0258]	144.2232
2b	[-0.4525, 0.8914, 0.0258]	47.8521
3a	[0.395, -0.0935, -0.9139]	115.6897
3b	[-0.395, 0.0935, 0.9139]	54.2429



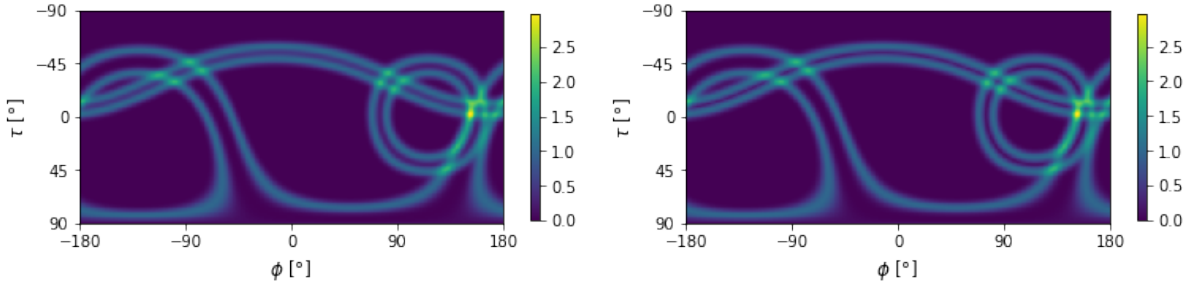
(a) Simple addition, $P_c = [153°, -3°]$ (b) Twin addition, $P_c = [153°, -3°]$

Figure 8.13: Sum of back-projections of correct cones using (a) simple addition and (b) twin addition



(a) Simple addition, $P_c = [164°, -12°]$ (b) Twin addition, $P_c = [164°, -12°]$

Figure 8.14: Sum of back-projections of incorrect cones using (a) simple addition and (b) twin addition



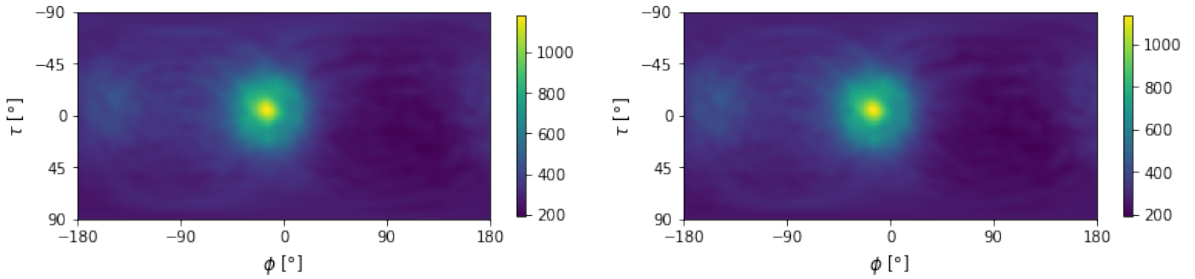
(a) Simple addition, $P_c = [153°, -3°]$

(b) Twin addition, $P_c = [153°, -3°]$

Figure 8.15: Sum of back-projections of correct and incorrect cones using (a) simple addition and (b) twin addition

8.3.4 Example with real data

The previous example used few and artificial data. There were 3 frames which resulted in 6 cones. This example uses the measurement with $P_m = [-15.3°, -4.95°]$ and $R_e = 30$ keV. The measurement has 3282 usable frames which results in 6069 cones. Figure 8.16 compares the sum of back-projections using simple addition in 8.16a with $P_c = [-14°, -5°]$ and twin addition in 8.16b with $P_c = [-15°, -5°]$. While the two figures might seem identical, the values for P_c illustrate that there is a small but significant difference, which results in twin addition having a slightly better result because simple addition $\varepsilon_a = 1.296°$ and for twin addition $\varepsilon_a = 0.303°$.



(a) Simple addition, $P_c = [-14°, -5°]$

(b) Twin addition, $P_c = [-15°, -5°]$

Figure 8.16: Sum of back-projections of a real measurement with $P_m = [-15.3°, -4.95°]$ using (a) simple addition and (b) twin addition

8.4 Multiplication of back-projections

In 8.3 all the different back-projections are added up. That can be translated as looking for a point that is located on this cone **or** that cone **or** that cone. P_c should actually be the point that is located on this cone **and** that cone **and** that cone. That would result in multiplication instead of addition.

8.4.1 Simple multiplication

Simply making a back-projection for every cone and multiplying them instead of adding them, would not give a usable result. There are two reasons: incorrect cones and small

numbers. The first problem will be explained and solved in the following paragraphs. The second problem will be solved in section 8.4.3.

Incorrect cones

The creation of a 2D map results in every point having a value between 0 and 1. However, the vast majority of the points will have a value of 0. Making a separate 2D map for every single cone also means that every single incorrect twin cone will get its own map. Because P_c is probably a 0 on that map — or at least at one of the maps of incorrect twin cones — the outcome for that point will always be 0, because at some point it gets multiplied by zero. This method results in a figure where every value is 0. Figure 8.17 illustrates this. The cones from table 8.1 were used for this figure.

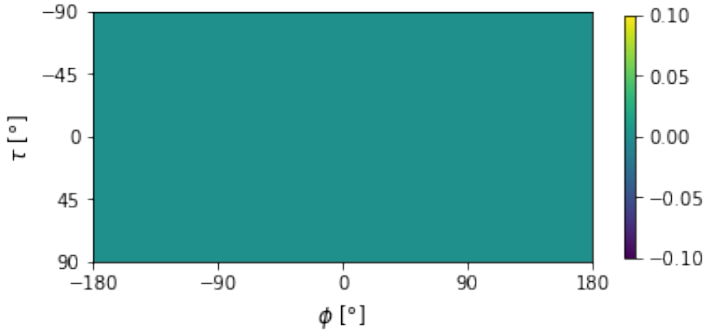


Figure 8.17: Simple multiplication of 2D maps of the cones defined in table 8.1

8.4.2 Twin multiplication

The problem about the back-projections of incorrect twin cones can be solved by using the same method as in twin addition. By keeping the twin cones together and taking the maximum value of the back-projections for each point to create the map, there should be no map where P_c is not located on the back-projection. Therefore, multiplication of the created 2D maps should result in a 2D map where the value for P_c is not 0. For the cones defined in table 8.1, this results in figure 8.18.

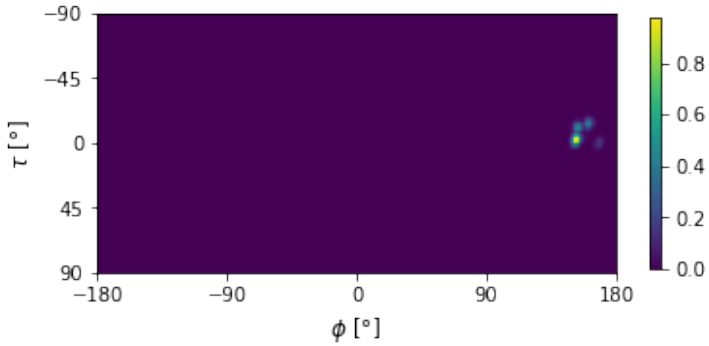


Figure 8.18: Twin multiplication of 2D maps of the cones defined in table 8.1, $P_c = [153°, -3°]$

However, this figure was made with few cones and artificial data. Applying the same principle on real data, does not yield usable results. Figure 8.19, made with the axes

and angles of the measurement with P_m = [-5.40°,-4.95°] and R_e = 30 keV, illustrates the steps taken when doing twin multiplication. Subfigure (1) illustrates the back-projection for the first usable frame. Subfigure (2) visualises the multiplication of the (1) with the back-projection of the second usable frame. Subfigure (3) visualises the multiplication of (2) with the back-projection of the third usable frame. This process repeats until there are 15 figures. Subfigures (1) to (3) illustrate the progression that might be expected. The progression from (3) to (4) is not what is expected. P_c of (4) is not close to P_c of subfigure (3). Taking a closer look at the legend also reveals that the maximum value is significantly lower. This means that the back-projection of the cone of usable frame 4 is probably not close to the previous back-projections. This results in a very low maximum value. The same thing — the maximum value lowering by a significant amount — happens again in other subfigures. This goes on until the maximum value becomes so low that Python cannot handle it. At that moment every point gets the value 0 and P_c = [-180°,90°]. To solve this problem, very small numbers have to be avoided. A solution is presented in section 8.4.3.

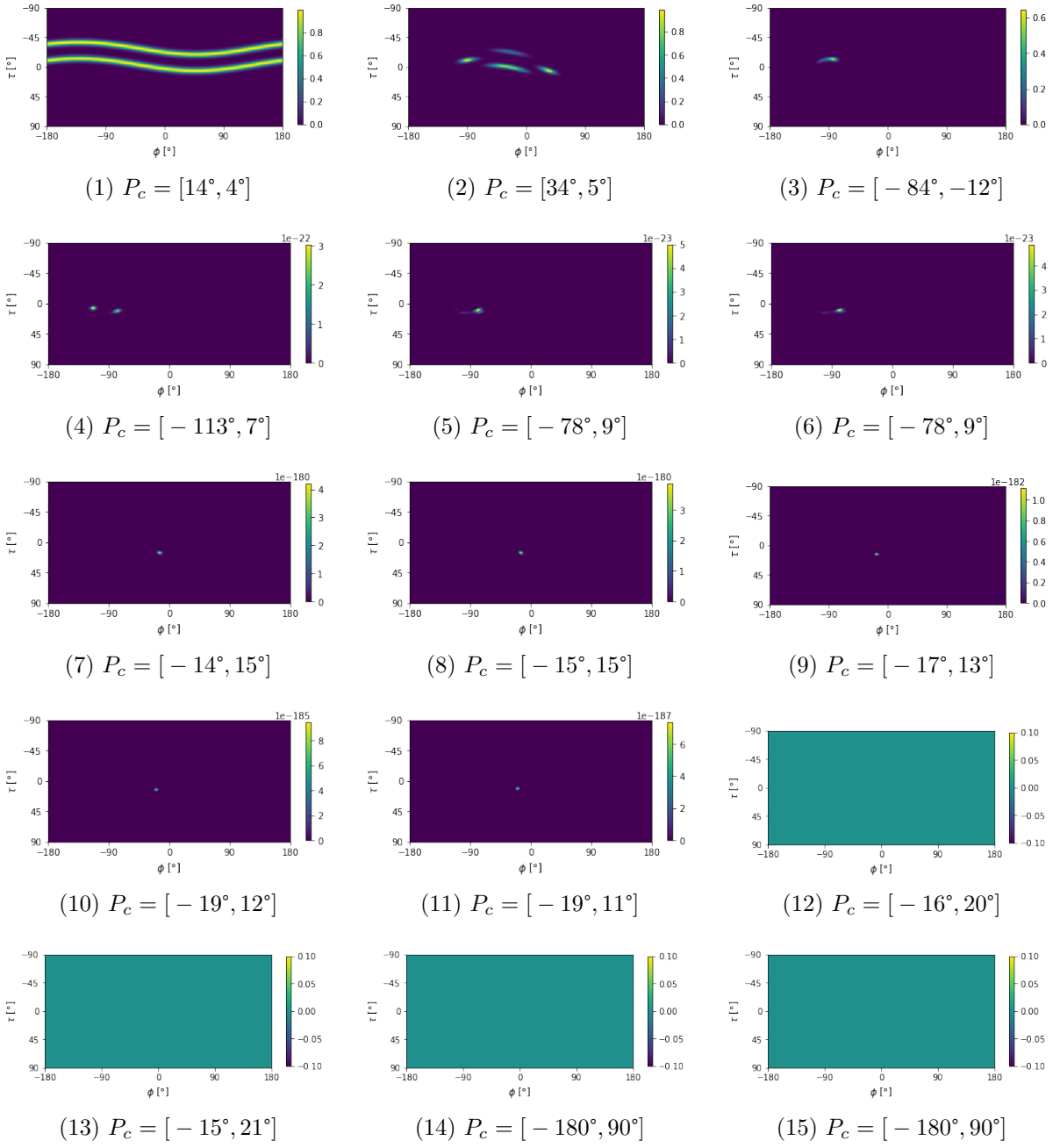


Figure 8.19: Step-by-step illustration of twin multiplication, $P_m = [-5.4°, -4.95°]$

8.4.3 Adjusting the creation of a single back-projection

In order to solve the problem explained in the previous paragraph, the original method for making a single back-projection (see section 8.2) must be changed. That new method will be referred to as δ_m. The original method resulted in values in range]0,1]. That meant each multiplication ended in a smaller value — except for 1, then the value remained the same. To avoid the value getting smaller each time, it is necessary that the absolute value of δ_m is greater than 1 (most of the time). There are different ways to solve that problem. A simple method could be to just do $\delta_m = \delta_g + 1$. That meets the requirement of —δ_m— being greater than 1, but upon testing it does not work because it quickly results in the values being to big (and thus becoming inf in Python) because of the multiplication of all the 2D maps. For the number of usable frames in our measurements, this could be solved

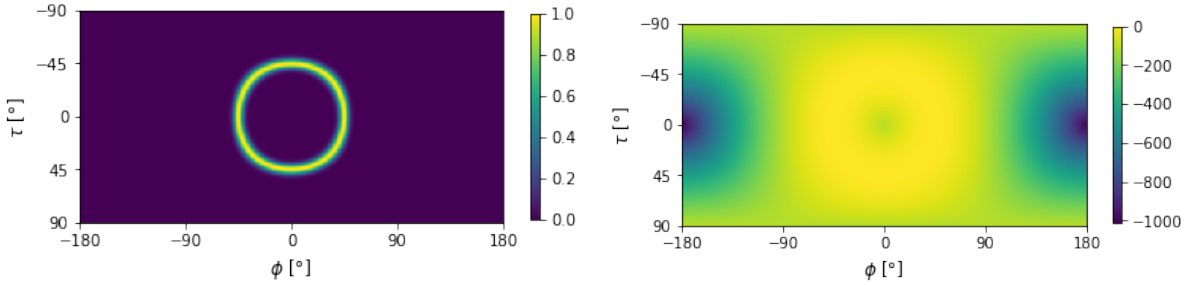
by doing $\delta_m = 1 + \frac{\delta_g}{100}$.

However, a better option is to take the natural logarithm of δ_g (see equation 8.13). Besides producing numbers whose absolute value is greater than 1 (in most cases), it provides two other advantages. First, the natural logarithm is the inverse of the exponential function, which means they cancel each other out (see equation 8.14). Second, by taking the natural logarithm the multiplication of 2D maps changes into addition because $\ln(a \cdot b) = \ln(a) + \ln(b)$.

$$\delta_m = \ln(\delta_g) = \ln(e^{\frac{-|\delta_{axis} - \theta|^2}{2 \cdot \sigma^2}}) \tag{8.13}$$

$$\delta_m = \frac{-|\delta_{axis} - \theta|^2}{2 \cdot \sigma^2} \tag{8.14}$$

To illustrate the differences between the original method of making a single back-projection and the new method, the results of the two methods are put side by side in figure 8.20. Figure 8.20a gives the result of a back-projection using equation 8.9, which is the same figure as figure 8.10. Figure 8.20b gives the result of a back-projection using equation 8.14.



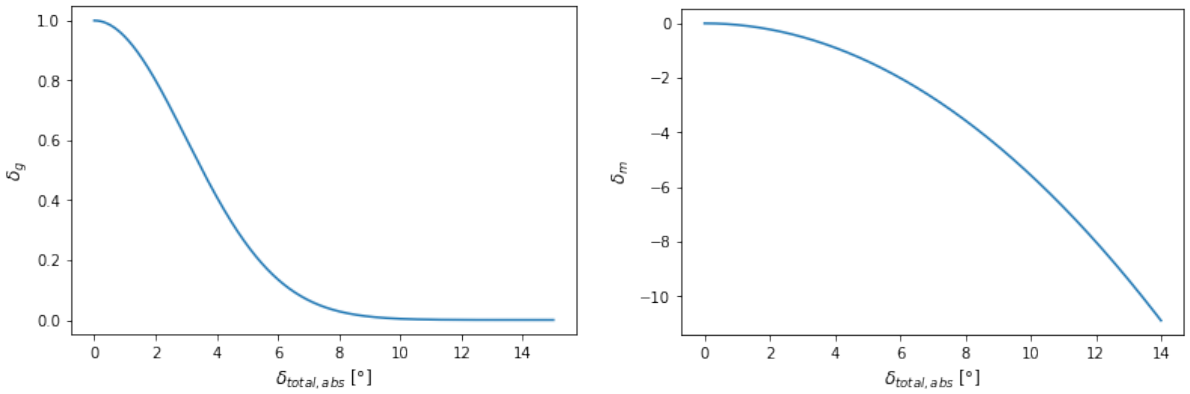
(a) (b)

Figure 8.20: The difference between calculating a 2D map for a single back-projection with $\vec{u}$ = [1,0,0], $\theta = 45°$ and $\sigma = 3$ using (a) δ_g and (b) δ_m

The back-projection in 8.20a is very clear, while the back-projection in 8.20b is still visible, but significantly less clear. Putting their gradients side by side (8.21) explains this observation. The gradient of 8.21a is in the range]0,1] and thus has a 'natural' cut-off value. The gradient of 8.21b is in the range]-∞,0] and does not have a natural (or imposed) cut-off value.

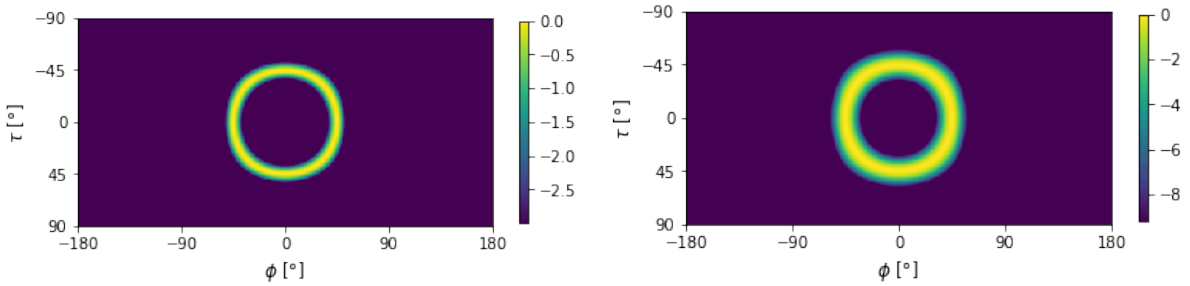
To improve the clarity of a single back-projection using δ_m, a cut-off value, referred to as v_c can be imposed. Equation 8.15 defines δ_m with a cut-off value, referred to as $\delta_{m,c}$. Figure 8.22 displays the result of $\delta_{m,c}$ with $\sigma = 3$ for $v_c = 0.05$ (figure 8.22a) and $v_c = 0.0001$ (figure 8.22b) and in doing so illustrates the impact of v_c. v_c here determines the width of the back-projection in a very similar way as σ did in 8.2.

$$\begin{cases} \delta_{m,c} = \delta_m & \text{if } \delta_m \leq \ln(v_c) \\ \delta_{m,c} = \ln(v_c) & \text{if } \delta_m \geq \ln(v_c) \end{cases} \tag{8.15}$$



(a) (b)

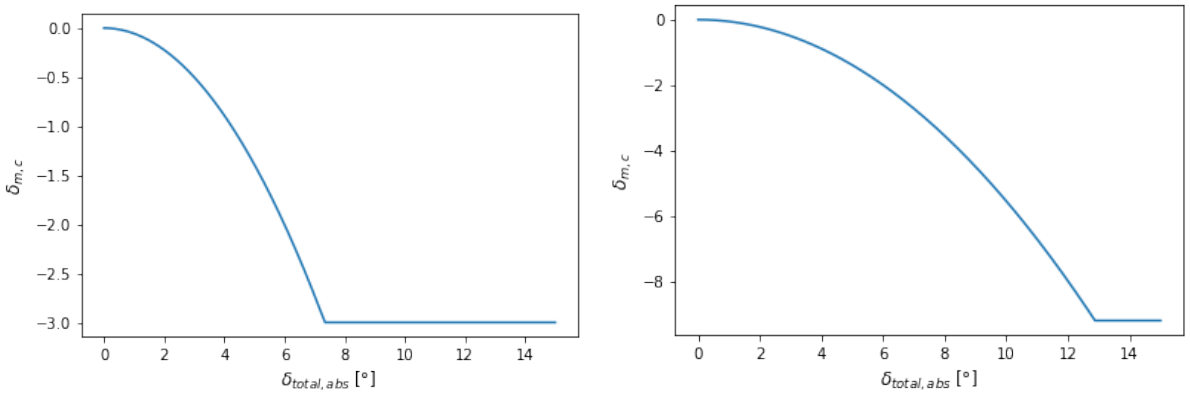
Figure 8.21: Difference in descent for (a) δ_g with $\sigma = 3$ and (b) δ_m with $\sigma = 3$



(a) (b)

Figure 8.22: 2D map of $\delta_{m,c}$ for a cone with $\vec{u} = [1,0,0]$, $\theta = 45°$, $\sigma = 3°$ and (a) $c_v = 0.05$ (b) $c_v = 0.0001$

Figure 8.23 displays the gradients of $\delta_{m,c}$ in function of $\delta_{total,abs}$ for different values of c_v. Figure 8.23a is the gradient that belongs to 8.22a where $v_c = 0.05$ and Figure 8.23b is the gradient that belongs to 8.22b where $v_c = 0.0001$. These gradients can be confirmed by simple calculations: ln(0.05) = -3.0 and ln(0.0001) = -9.2.



(a) (b)

Figure 8.23: Difference in descent for $\delta_{m,c}$ with $\sigma = 3$ and (a) $c_v = 0.05$ (b) $c_v = 0.0001$

Example with real data

This example uses the same measurement as for the example of addition, $P_m = [-15.3°, -4.95°]$ and $R_e = 30$ keV. Figure 8.24 visualises the result for $\sigma = 3$ and $v_c = 0$. This is the same as not applying a cut-off value because $\ln(0) = -\infty$. Figure 8.25 visualises the result for $\sigma = 3$ and (a) $v_c = 0.05$, (b) $v_c = 0.0001$. These figures illustrates that using a cut-off value results in better accuracy because $\varepsilon_a = 3.424°$ for $v_c = 0$ and $\varepsilon_a = 0.303°$ for $v_c = 0.05$ and $v_c = 0.0001$. Chapter 9 researches what a good value is for v_c

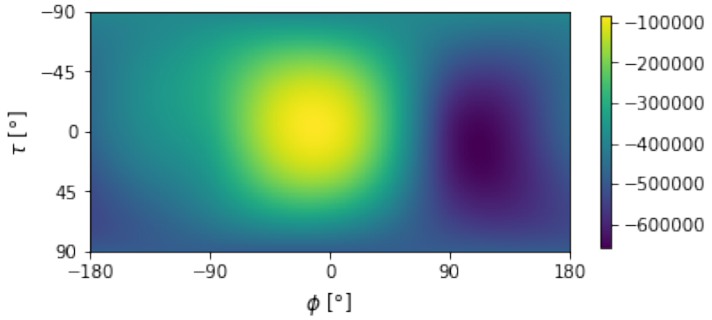
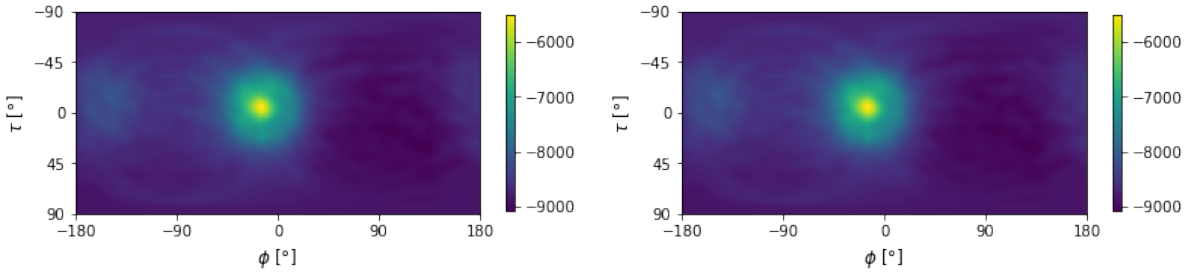


Figure 8.24: Multiplication of back-projections of a real measurement with P_m = [-15.3°,-4.95°] with $\sigma = 3$ and $v_c = 0$ that results in $P_c = [-12°, -4°]$



(a) (b)

Figure 8.25: Multiplication of back-projections of a real measurement with P_m = [-15.3°,-4.95°] with $\sigma = 3°$ and (a) $v_c = 0.05$ that results in $P_c = [-15°, -5°]$, (b) $v_c = 0.001$ that results in $P_c = [-15°, -5°]$

Twin multiplication with the use of a cut-off value will from now on be referred to as simply 'multiplication'.

8.5 Zooming in to decrease computing time

To have an idea about the computation time for the different methods, a comparison was performed. For every method (simple addition with $R_e = 30$ keV and $\sigma = 3°$, twin addition with $R_e = 30$ keV and $\sigma = 3°$ and multiplication with $R_e = 30$ keV, $\sigma = 3°$ and $v_c = 0.05$) the processing time was calculated for the measurement with P_m = [-15.3°,-4.95°] It is important to note that the process time only measured the time to actually calculate P_c, starting with the angles, axes and unravelled sphere already calculated. In order to get accurate time readings, the function time.process_time() is used as to not include the time the computer is busy doing other tasks. A first time doing these calculations, which

existed of calculating the processing time 5 different times and then taking the average, resulted in the average time being 98 s for simple addition, 98 s for twin addition and 99 s for multiplication. On a different day, this resulted in the average time being 78 s for simple addition, 79 s for twin addition and 78 s for multiplication. This shows that the calculation time is still heavily subject to fluctuations even though the function time.process_time() was used. The fluctuations are due to the fact that timing can never really be guaranteed in a real computer. That is because the state of the computer is never the same. Many things can differ: certain bytes still being in the cache, the python script getting 2 x 15 ms instead of 1 x 30 ms, the temperature of the CPU being too high, ... That means that the exact calculation times are not that important but that the ratio of the calculation times calculated at roughly the same moment is important. Therefore, it can be concluded that the difference in computing time between the three different implementations is insignificant. To speed up the process of calculating P_c, a zoom function is implemented. The sections below explain the implementation and give an example. The results can be found in chapter 9.

8.5.1 Implementation

Zooming starts with $a_S = a_{S,orig}$ and ends with $a_S = a_{S,new}$. It is implemented by first doing a back-projection with a sphere with accuracy $a_{S,orig}$ which results in a figure of $(\frac{180}{a_{S,orig}} + 1)$ by $(\frac{360}{a_{S,orig}} + 1)$. Then P_c is calculated for that figure. The next step is making part of a sphere that is located around P_c as a raster. This square raster has a size of $(2 \cdot \frac{a_{S,orig}}{a_{S,new}} + 1)$ by $2(\cdot\frac{a_{S,orig}}{a_{S,new}} + 1)$. For every point in the raster, a back-projection is done. Figure 8.26 illustrates the concept with $a_{S,orig} = 5°$ and $a_{S,new} = 1°$. The numbers in blue in the figure are ϕ_{fig} and τ_{fig}. The numbers in black are ϕ_{diff} and τ_{diff}. Subsequently, P_c is calculated using equation 8.17 which in turn uses equation 8.16.

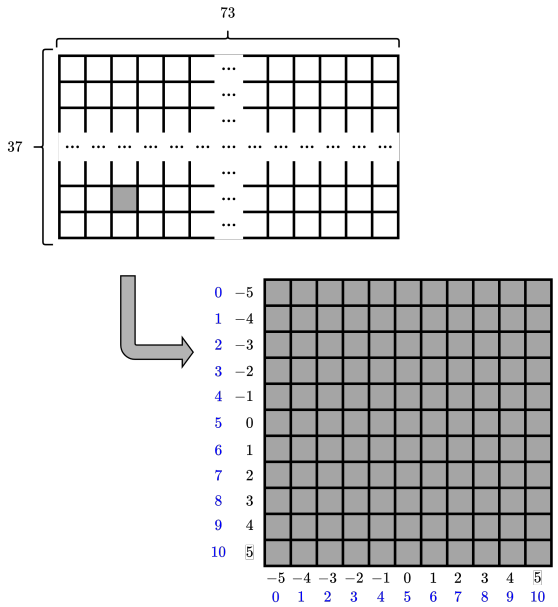


Figure 8.26: Illustration of the zoom function

$$\begin{cases} \phi_{diff} = \left(\phi_{fig} - \frac{a_{S,orig}}{a_{S,new}}\right) \cdot a_{S,new} \\ \tau_{diff} = \left(\tau_{fig} - \frac{a_{S,orig}}{a_{S,new}}\right) \cdot a_{S,new} \end{cases} \tag{8.16}$$

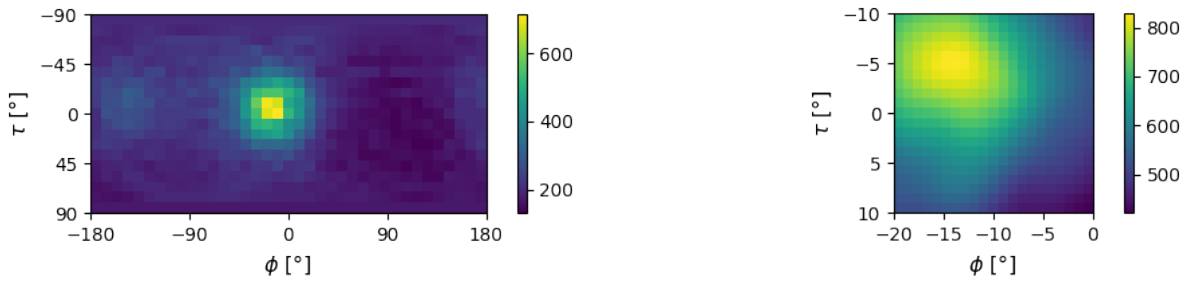
$$\begin{cases} \phi_{new} = \phi_c - \phi_{diff} \\ \tau_{new} = \tau_c - \tau_{diff} \end{cases} \tag{8.17}$$

Two things should be noted. First, it is possible to zoom in more than once and make it an iterative process. Second, there is no loss of accuracy when results are calculated using the zoom implementation compared to immediately calculating the results with an accuracy of $a_{S,new}$.

8.5.2 Example

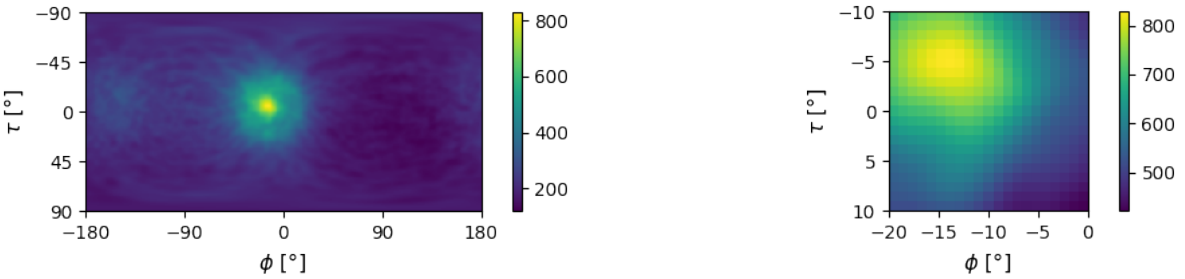
This subsection uses an example to illustrate the results of the previous subsection. Like in the previous subsection, back-projection with simple addition is used. The measurement with $P_m = [-15.3°, -4.95°]$ is used for this example with $R_e = 30$ keV and $\sigma = 2°$.

On the one hand, a back-projection with $a_S = 10°$ is made. This results in figure 8.27a, then the zoom function is used to bring the accuracy to $a_S = 1°$. This results in figure 8.27b. On the other hand, a back-projection with an angle accuracy of 1° is made. This results in figure 8.28a. Figure 8.28b literally zooms in to display only part of figure 8.28a. Comparing figure 8.28b to figure 8.27b illustrates that starting with a lower accuracy (which means a higher value for a_S) and then using the zoom function results in exactly the same figure as literally zooming in on the figure made with the higher accuracy (which means a lower value for a_S).



(a) (b)

Figure 8.27: Illustration of the zoom function applied on a real measurement starting from (a) $a_S = 10°$ to (b) $a_S = 1°$



(a) (b)

Figure 8.28: Illustration of (a) the back-projection of a real measurement using simple addition with $a_S = 1°$ and (b) a part of that same back-projection zoomed in on the area around P_c

Chapter 9

Results and discussion

This chapter discusses the achieved results using the methods from chapter 8. For all the methods there were certain parameters that can be varied. This chapter gives an overview of the different results. For the tables that discuss the angular error ε_a, there are always two tables: one has the values of the mean angular error $\overline{\varepsilon_a}$ while the other has the values of the root-mean-square angular error RMS(ε_a). While the tables with $\overline{\varepsilon_a}$ give the parameter values that give the lowest $\overline{\varepsilon_a}$, the tables with RMS(ε_a) are also important because they filter out the bigger angular errors. That might be important if small errors are not really a problem because there is an extra safety margin. However this safety margin might not be big enough to compensate for larger errors. There will be no exact results about what 'the best' values are for certain parameters is, because the choice for certain methods and parameter values can differ between different projects. Some projects might focus on obtaining the smallest angular error possible, while others might want to minimise measurement time or processing time. For every table in this chapter $a_S = 1°$. That means that even if everything worked perfectly and there were no uncertainties, there would still be an error because most of the values for ϕ_m and τ_m are not integers. If for every measurement ϕ_c and τ_c were the integers closest to ϕ_m and τ_m, then $\overline{\varepsilon_a}$ would be 0.290°.

9.1 Simple addition

The 2 parameters that can be varied for simple addition are R_e and σ. This results in tables 9.1 and 9.2. The best results are $\overline{\varepsilon_a} = 1.414°$ with $R_e = 30$ keV and $\sigma = 2°$ and RMS(ε_a) = 1.632° with $R_e = 35$ keV and $\sigma = 2°$.

Table 9.1: $\overline{\varepsilon_a}$ for different values of σ and R_e for simple addition

$\overline{\varepsilon_a}$ [°]		R_e [keV]				
		25	**30**	**35**	**40**	**45**
σ [°]	**1**	1.930	1.825	1.780	1.836	1.818
	2	1.518	1.414	1.419	1.450	1.477
	3	1.532	1.534	1.593	1.608	1.663
	4	1.726	1.756	1.855		
	5	1.950	2.042	2.122		

Table 9.2: RMS(ε_a) for different values of σ and R_e for simple addition

RMS(ε_a) [°]		R_e [keV]				
		25	**30**	**35**	**40**	**45**
σ [°]	**1**	2.211	2.145	2.022	2.128	2.079
	2	1.767	1.653	1.632	1.658	1.733
	3	1.759	1.762	1.823	1.833	1.918
	4	2.003	2.039	2.150		
	5	2.290	2.404	2.502		

9.2 Twin addition

The 2 parameters that can be varied for twin addition are R_e and σ. This results in tables 9.3 and 9.4. The best results are $\overline{\varepsilon_a} = 1.340°$ with $R_e = 30$ keV and $\sigma = 3°$ and RMS(ε_a) $= 1.537°$ with $R_e = 30$ keV and $\sigma = 3°$.

Table 9.3: $\overline{\varepsilon_a}$ for different values of σ and R_e for twin addition

$\overline{\varepsilon_a}$ [°]		R_e [keV]			
		25	**30**	**35**	**40**
σ [°]	**1**	1.873	1.746	1.787	1.765
	2	1.428	1.365	1.373	1.369
	3	1.378	1.340	1.351	1.416
	4	1.462	1.476	1.559	

Table 9.4: RMS(ε_a) for different values of σ and R_e for twin addition

RMS(ε_a) [°]		R_e [keV]			
		25	**30**	**35**	**40**
σ [°]	**1**	2.138	2.019	2.043	2.032
	2	1.640	1.557	1.562	1.563
	3	1.577	1.537	1.553	1.606
	4	1.688	1.702	1.786	

9.3 Multiplication

The 3 parameters can be varied for multiplication are R_e, σ and v_c. The first calculations vary R_e, σ and have a fixed $v_c = 0.05$. This results in tables 9.5 and 9.6. The best results are $\overline{\varepsilon_a} = 1.332°$ with $R_e = 30$ keV and $\sigma = 2°$ and RMS(ε_a) $= 1.520°$ with $R_e = 30$ keV and $\sigma = 2°$.

Table 9.5: $\overline{\varepsilon_a}$ for different values of σ and R_e with $v_c = 0.05$ for multiplication

$\overline{\varepsilon_a}$ [°]		R_e [keV]			
		25	**30**	**35**	**40**
σ [°]	**1**	1.768	1.670	1.664	1.705
	2	1.424	1.332	1.339	1.374
	3	1.501	1.533	1.583	1.615

Table 9.6: RMS(ε_a) for different values of σ and R_e with $v_c = 0.05$ for multiplication

RMS(ε_a) [°]	R_e [keV]				
		25	**30**	**35**	**40**
σ [°]	**1**	2.020	1.906	1.819	1.969
	2	1.613	1.520	1.542	1.560
	3	1.728	1.764	1.812	1.847

The next parameter that needs to be inspected is v_c. Because $\overline{\varepsilon_a}$ and RMS(ε_a) for $R_e = 30$ and $R_e = 35$ with $\sigma = 2$ where close to each other, the choice is made to calculate the $\overline{\varepsilon_a}$ and RMS(ε_a) for $R_e = 30$ keV and $R_e = 35$ keV with varying values for parameter v_c and fixed $\sigma = 2$. This results in tables 9.7 and 9.8. The best results are $\overline{\varepsilon_a} = 1.332°$ with $R_e = 30$ keV and $v_c = 0.05$ and RMS(ε_a) = 1.520° with $R_e = 30$ keV and $v_c = 0.05$. Combining the results from table 9.5, 9.6, 9.7 and 9.8 leads to the parameter values being $R_e = 30$ keV, $\sigma = 2°$ and $v_c = 0.05$ to get the best results for $\overline{\varepsilon_a}$ and RMS(ε_a).

Table 9.7: $\overline{\varepsilon_a}$ for different values of R_e and v_c with $\sigma = 2°$ for multiplication

$\overline{\varepsilon_a}$ [°]	R_e [keV]		
		30	**35**
v_c	**0**	11.044	11.238
	0.0001	1.700	1.724
	0.01	1.421	1.405
	0.03	1.370	1.333
	0.05	1.332	1.339
	0.07	1.370	1.357
	0.1	1.396	1.344

Table 9.8: RMS(ε_a) for different values of R_e and v_c with $\sigma = 2°$ for multiplication

RMS(ε_a) [°]	R_e [keV]		
		30	**35**
v_c	**0**	11.938	12.095
	0.0001	1.934	1.992
	0.01	1.610	1.589
	0.03	1.562	1.531
	0.05	1.520	1.542
	0.07	1.559	1.557
	0.1	1.588	1.534

9.4 Validating the values for R_e

Figure 9.1 displays the energy spectrum of the measurement with $P_m = [-20.25°, -50.40°]$. The histogram is in the range [10, 1000] and has 300 bins. Number 1 is the XRF peak (see 5.2.3), number 2 is the Pb X-ray peak, number 3 is the back-scatter peak, number 4 is the Compton edge and number 5 is where full absorption of ^{137}Cs takes place. The important part for this research is number 5.

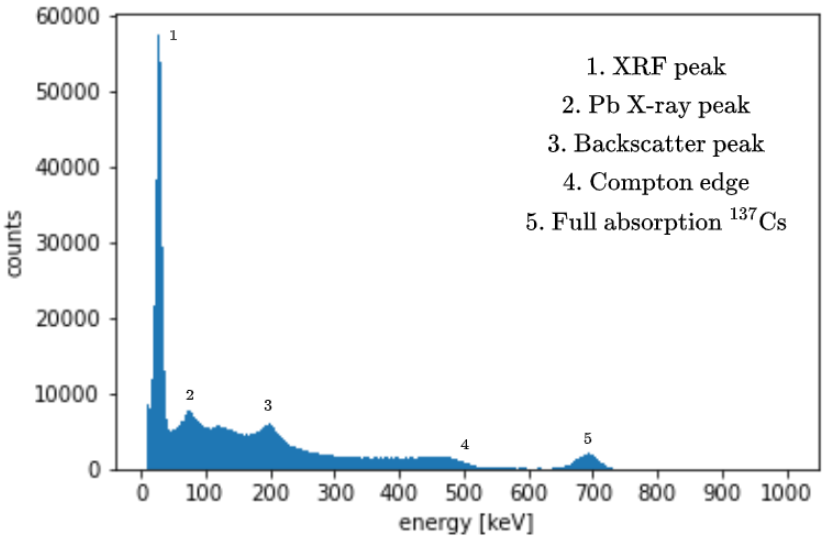


Figure 9.1: Energy spectrum of a real measurement with the different peaks and edges indicated

Figure 9.2 zooms in on the last peak. This is the part of the spectrum where the Compton interactions can be found. The peak is located around 700 keV, therefore all the frames with an energy between 600 keV and 800 keV are used to make this figure. Calculating the mean and standard deviation results in mean = 686.7 keV and standard deviation = 24.3 keV. The full width at half maximum (FWHM) is then calculated using equation 9.1. This results in FWHM = 57.3 keV. This value confirms that 30 keV (or 35 keV) is an acceptable value for R_e.

$$\text{FWHM [keV]} = 2 \cdot \sigma[\text{keV}] \cdot \sqrt{2 \cdot \ln 2} \tag{9.1}$$

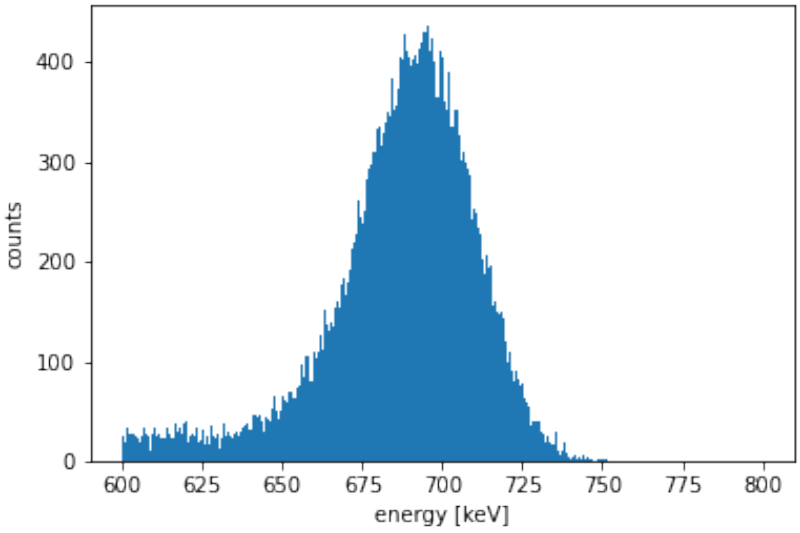


Figure 9.2: Energy spectrum of a real measurement zoomed in on the peak where full absorption of ^{137}Cs takes place

9.5 Zoom

The zoom function, explained in 8.5, was tested using the measurement with $P_m = [-15.3°, -4.95°]$ using simple addition with $R_e = 30$ keV and $\sigma = 2°$. Table 9.9 lists the tested zoom sequences and their process time. The process time was calculated in the same way as in section 8.5. The process time for each zoom sequence was measured 4 times and then the average was calculated. As established in section 8.5, the exact times are not important but the ratio is.

Table 9.9: Process time for different zoom sequences

Zoom sequence	**Process time (s)**	**Ratio**
$1°$	73.3789	100
$5° \rightarrow 1°$	0.8867	1.21
$10° \rightarrow 1°$	0.5430	0.74
$20° \rightarrow 1°$	0.6289	0.86
$20° \rightarrow 5° \rightarrow 1°$	0.4922	0.67
$40° \rightarrow 5° \rightarrow 1°$	0.5391	0.73
$30° \rightarrow 3° \rightarrow 1°$	0.5234	0.71
$30° \rightarrow 5° \rightarrow 1°$	0.4531	0.62
$60° \rightarrow 20° \rightarrow 5° \rightarrow 1°$	0.5742	0.78
$90° \rightarrow 30° \rightarrow 10° \rightarrow 1°$	0.6484	0.88

The table shows that the sequence $30° \rightarrow 5° \rightarrow 1°$ is the fastest one. It is about 160 times faster than not using the zoom function. The findings from table 9.9 can be explained using some calculations. When calculating P_c most of the computation time is spent on calculating the back-projections which includes a matrix multiplication. That is a rather costly calculation, which means it is better if those matrices are smaller.

Directly calculating P_c with $a_S = 1°$ leads to a matrix multiplication of $\vec{u}$ with a matrix of 3 by **65341** (181 x 361) for every cone. When using a zoom function, there are more matrix multiplications, but the numbers are significantly smaller. Starting with $a_S = 5°$ leads to a matrix multiplication of $\vec{u}$ with a matrix of 3 by **2701** (37 x 73) for every cone. This is then followed by a matrix multiplication of $\vec{u}$ with a matrix of 3 by **121** (11 x 11) for every cone. When using the iterative zoom $30° \rightarrow 5° \rightarrow 1°$, the numbers become **91** (13 x 7) $\rightarrow$ **169** (13 x 13) $\rightarrow$ **121** (11 x 11).

Using more than 3 steps in the iteration brings the size of the matrices further down, but not the process time. That is because those matrix multiplications are not the only calculations that need to be executed. In each iterative step a number of other things also has to be calculated (for example computing ϕ_{diff} and τ_{diff}), which means that more iterative steps do not always lead to a shorter process time.

9.6 $\overline{\varepsilon_a}$ in function of number of Compton interactions

9.6.1 Implemented calculations

All the tables in this chapter used the whole measurement to calculate the values for $\overline{\varepsilon_a}$ and RMS(ε_a). To have an idea about the amount of data necessary to get decent results

for ε_a — where the interpretation of decent depends on the project — other calculations need to be performed. Plotting a graph of $\overline{\varepsilon_a}$ in function of the number of Compton events — which is the same as the number of usable frames — can lead to new insights. The steps taken to plot such a curve are explained in the paragraphs below. Figure 9.3 is added to keep track of the different steps in the explanation.

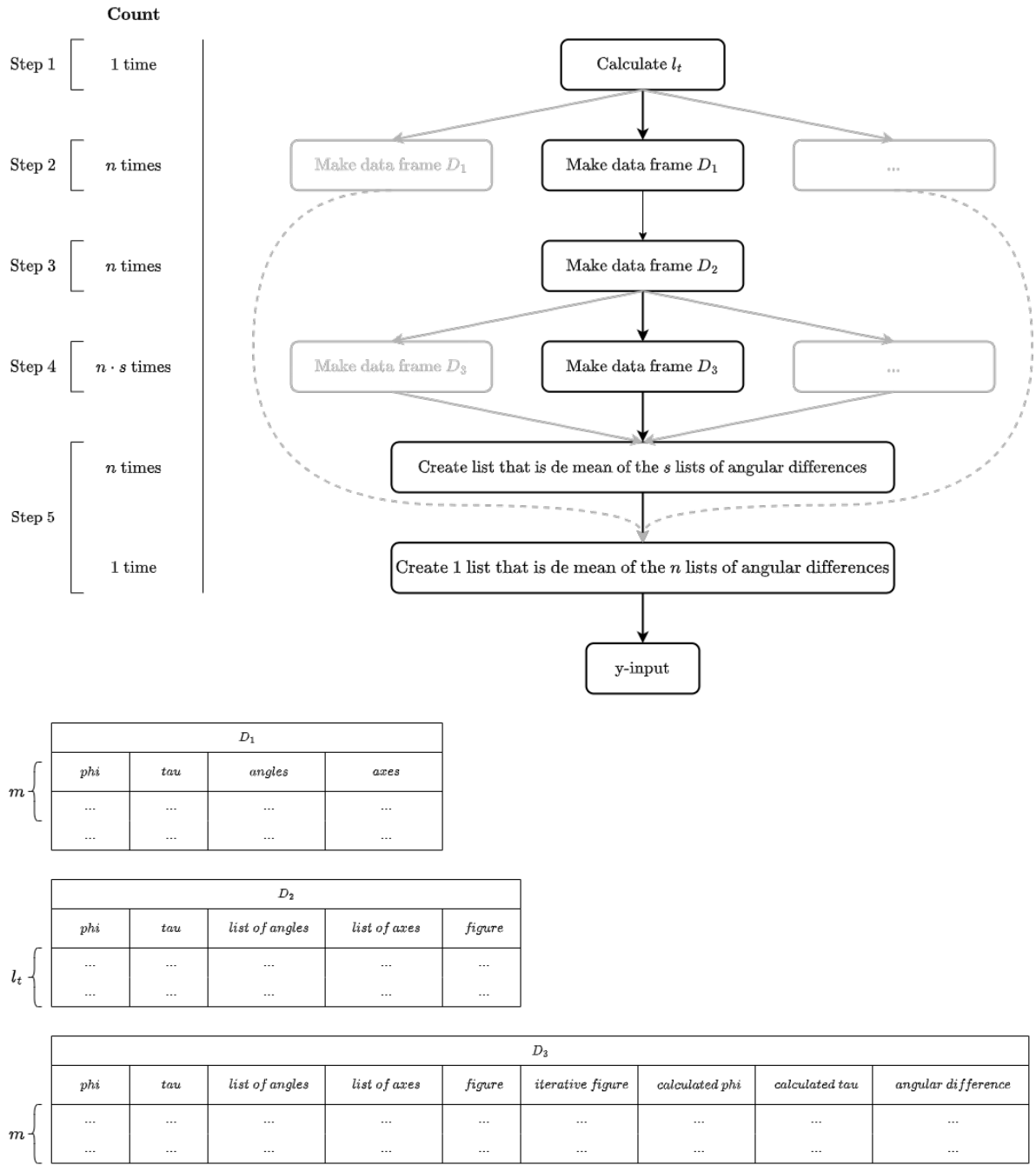


Figure 9.3: Visual support to help understand the creation of the y-input

Step 1: Determine maximum number of Compton events to use in the graph

To determine the maximum number of Compton events that can be used in the graph, it is necessary to find the minimum number of usable frames for all n measurements with a certain value for R_e. The lowest value for R_e that will be used in the following

calculation is $R_e = 30$ keV, because a smaller value for R_e leads to a lower number of usable frames, the lowest number of usable frames for $R_e = 30$ keV is determined. That minimum number is 2684, which is rounded down to a total length $l_t = 2500$.

Step 2: Create n shuffled data frames D_1 with l_t rows

A data frame D_1 with columns *phi*, *tau*, *angles* and *axes* is created for every single measurement. This data frame has as many rows as there are usable frames. The rows are then randomly shuffled. A seed is used so the results can be reproduced. The data frame is then cut short to a length l_t.

Step 3: Create a new data frame with slices D_2

A new data frame D_2 is created by slicing data frame D_1 into different slices. Each slice has a slice length l_s. This new data frame has m rows with $m = \frac{l_t}{l_s}$. Each row contains the columns *phi*, *tau*, *list of angles* and *list of axes*. The columns *list of angles* and *list of axes* contain l_s elements. A new column *figure* is created by doing a back-projection for the cones defined by the *list of angles* and *list of axes* of that row.

Step 4: Create s new data frames D_3

s new data frames D_3 are created. Each of those data frames is a shuffled version of D_2 with a different *random state*. That means that D_3 also contains the columns *phi*, *tau*, *list of angles*, *list of axes* and *figure*. New columns are added to this data frame. The first one is the column *iterative figure*. This column contains the sum of every *figure* of every row up to and including that row itself. That means that *iterative figure* of the first row (index 0) contains a figure made with l_s usable frames, the second row contains a figure made with 2 x l_s usable frames, the third row with 3 x l_s usable frames and so on. The last row contains a figure made with $m \cdot l_s = l_t$ usable frames. Then the columns *calculated* ϕ and *calculated* τ are added. Those columns contain ϕ_c and τ_c using the figure of the column *iterative figure*. A last column, *angular difference*, is then added that calculates the angular difference ε_a between P_m, defined by ϕ, τ, and P_c, defined by *calculated phi* and *calculated tau*.

Step 5: Combining the column *angular difference* of all data frames to create the y-input

There is now a total of $n \cdot s$ data frames that all have m rows. n lists with m values are created that take the mean of all the s values for that row. Then 1 list with m values is created that takes the mean of all the n values for that row. That list is the y-input.

Step 6: Create the x-input

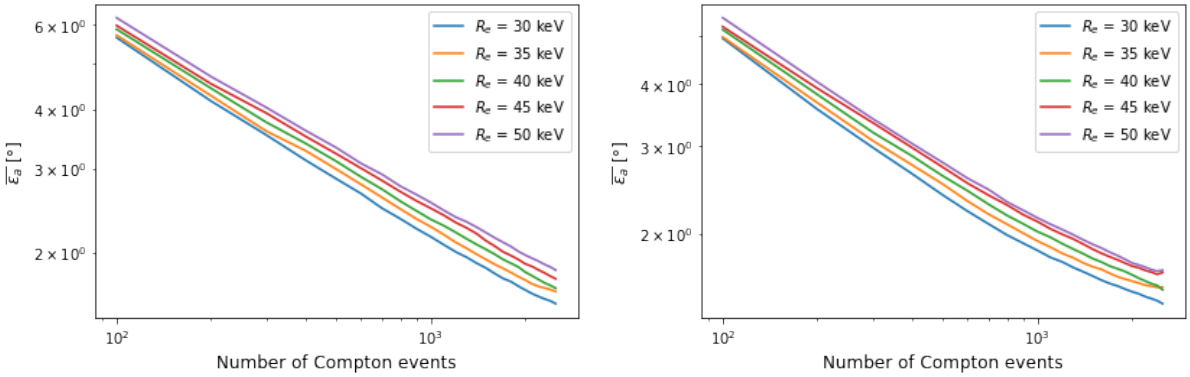
The x-input is a list that has the number of usable frames. The first element of that list is l_s, the second element is $2 \cdot l_s$, ..., the last element is $m \cdot l_s = l_t$.

9.6.2 Resulting figures

In this section different plots of $\overline{\varepsilon_a}$ in function of the number of Compton events are displayed. Only twin addition will be used because it keeps the cones of the usable frames together. Therefore twin addition will simply be referred to as 'addition' from now on. The previous sections of the chapter revealed that $R_e = 30$ keV and $R_e = 35$ keV were the two values that gave the best results. For the sake of experimental research $R_e = 40$ keV, $R_e = 45$ keV and R_e 50 keV are added to that list. These figures are made using the steps explained in section 9.6.1 with $n = 276$, $l_t = 2500$, $l_s = 100$, $m = 25$ and $s = 50$.

Addition

Figure 9.4 displays $\overline{\varepsilon_a}$ in function of the number of Compton events for R_e ranging from 30 keV to 50 keV and (a) $\sigma = 2°$ and (b) $\sigma = 3°$. These two figures display an exponential descent and illustrate that $\overline{\varepsilon_a}$ decreases when R_e decreases. This is logical because a smaller value for R_e means that the frames selected as usable frames have average energies closer to the ideal energy of E_γ. Figure 9.5 combines figures 9.4a and 9.4b for easy comparison. This figure illustrates that addition with $\sigma = 3°$ outperforms addition with $\sigma = 2°$. Table 9.3 verifies that.



(a) (b)

Figure 9.4: $\overline{\varepsilon_a}$ in function of the number of Compton interactions for addition for varying values of R_e and (a) $\sigma = 2°$ and (b) $\sigma = 3°$

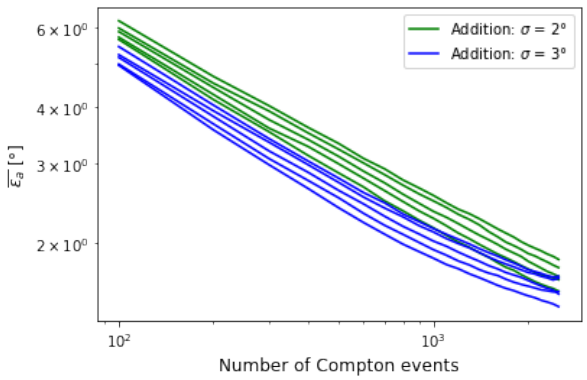


Figure 9.5: Comparison of $\overline{\varepsilon_a}$ in function of the number of Compton interactions for addition for $\sigma = 2°$ and $\sigma = 3°$

Multiplication

Figure 9.6 displays $\overline{\varepsilon_a}$ in function of the number of Compton events for multiplication with $\sigma = 2°$ and $v_c = 0.05$. In this figure $\overline{\varepsilon_a}$ also decreases when R_e decreases. That is logical for the same reason as for addition.

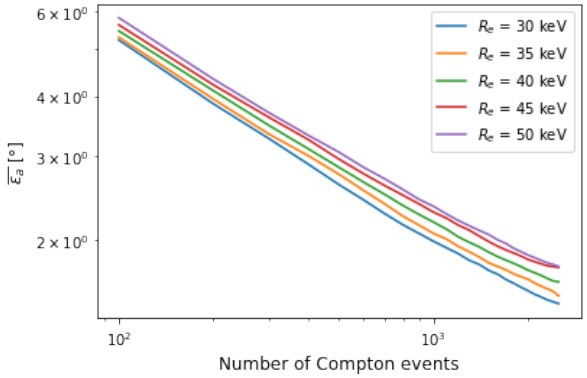


Figure 9.6: $\overline{\varepsilon_a}$ in function of the number of Compton interactions for multiplication with $\sigma = 2°$ and $v_c = 0.05$

Comparison

In figure 9.7 addition with $\sigma = 2°$, addition with $\sigma = 3°$ and multiplication with $\sigma = 2°$ and $v_c = 0.05$ is displayed. $R_e = 30$ keV is used for all three because that value gave the best results. The figure demonstrates that multiplication performs better than addition with $\sigma = 2°$ but worse than addition with $\sigma = 3°$.

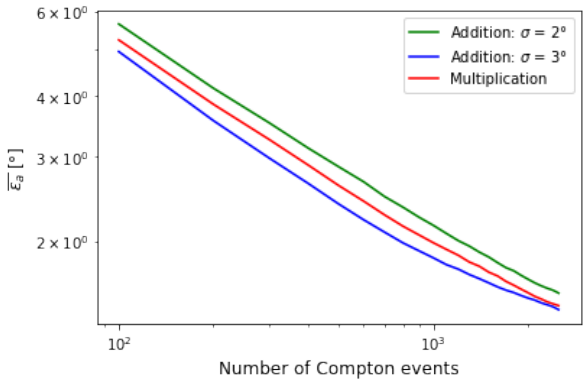
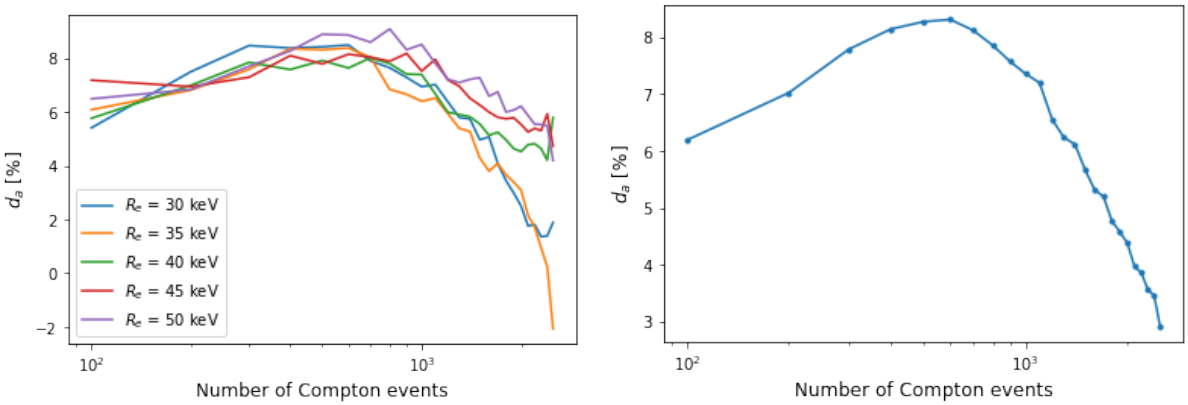


Figure 9.7: Comparison of $\overline{\varepsilon_a}$ in function of the number of Compton interactions with $R_e = 30$ keV for addition with $\sigma = 2°$, addition with $\sigma = 3°$ and multiplication with $\sigma = 2°$ and $v_c = 0.05$

It might seem like these results contradict the results from section 9.2 and section 9.3 because multiplication gave slightly more accurate results than twin addition. However the results in these tables used all the usable frames for every measurement. The curves in figure 9.7 used a limited number of usable frames. And while addition shows better results in figure 9.5, the difference between the two decreases when the number of Compton events increases. Figure 9.8, which uses a logarithmic scale for the x-axis and a linear scale for the y-axis, visualises the difference in accuracy of addition compared to multiplication, referred to as d_a, in function of the number of Compton events. Figure 9.8a displays the

curves for the different values of R_e and figure 9.8b displays the mean of the curves of figure 9.8a with the markers highlighting the exponential scale. The curves all show a downward trend starting at about 600 Compton events. The curve for $R_e = 35$ keV even crosses the threshold of 0%, which means that for $R_e = 35$ keV and 2500 Compton events multiplication is more accurate than addition.



(a) (b)

Figure 9.8: Difference in accuracy of addition compared to multiplication in function of the number of Compton events

9.6.3 Comparison with state-of-the-art

Comparison with the state-of-the-art means in this research comparing with [25].

An important aspect to take into consideration is the difference in Compton camera and environment management used in [25] compared to the one used in this thesis. In [25] the CdTe detector is equipped with a 1 mm thick crystal. It has 16 by 16 pixels with a pixel pitch of 625 μm leading to a total sensitive area of 1 cm^2. The operating temperature is a stable -10 °C obtained by using coolers. The detector of this thesis also has a 1 mm thick CdTe layer. However, it has 256 by 265 pixels with a pixel pitch of 55 μm, leading to a total sensitive area of 1.98 cm^2. The measurements are done at room temperature without temperature management which means the detector heats up during those measurements and has a less stable temperature. Therefore, while the explanation below often speaks of one algorithm versus another algorithm, the comparisons are not a direct comparison between algorithms, but are actually comparisons between the combination of a certain technologies combined with these algorithms.

The data from [25] is presented in appendix B and is used to make figure 9.9. This figure compares the back-projection implementation of this thesis with addition ($R_e = 30$ keV, $\sigma = 3°$), the back-projection implementation of this thesis with multiplication ($R_e = 30$ keV, $\sigma = 2°$, $v_c = 0.05$), the back-projection implementation from [25] and the neural network from [25].

As in the previous figures, this figure displays the exponential descent of the results calculated using the methods created in this thesis. When comparing the methods implemented in this thesis with the methods from [25], the figure demonstrates that both

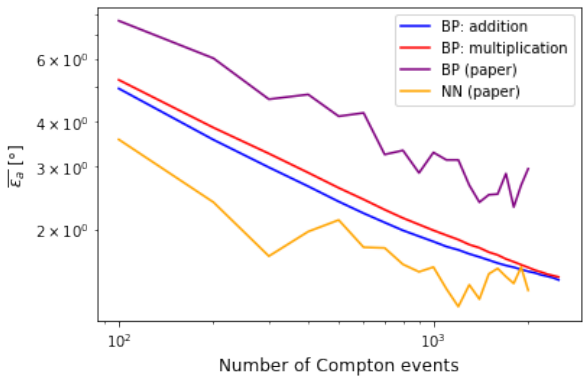


Figure 9.9: Comparison of angular errors $\overline{\varepsilon_a}$ in function of number of Compton events for the different implementations in this thesis and the implementations in [25]

back-projection algorithms implemented in this thesis outperform the back-projection from [25], but do not outperform the neural network from [25].

To have a better idea about the data, certain things can be replaced or added. The first option is to replace the mean with a simplified version of a box plot. For every x-value, there are 5 y-values plotted that represent 5 different things: the minimum, the first quartile, the median, the third quartile and the maximum. This illustrates the distribution. Figure 9.10 visualises the results. A second option is to add an error bar with a length of $2 \cdot std$ (one above the mean and one below). std is calculated using equation 9.2. This also illustrates the distribution, but implies that that distribution is Gaussian. This results in figure 9.11. A third option is to take the standard error of the mean (SEM) into account and add an error bar with a length of $2 \cdot$SEM. SEM is calculated using equation 9.3. This results in figure 9.12. This could be used to fit a curve. When fitting a curve without calculating SEM, every point will be treated as equally correct. However, when SEM is calculated, a curve can be fitted where the points with a smaller SEM have more weight than points with a larger SEM. This leads to a more accurate fit.

$$std = \sqrt{\frac{1}{s}\sum_{i=1}^{s}\overline{\varepsilon_a^2}} \tag{9.2}$$

$$\text{SEM} = \frac{std}{\sqrt{s-1}} \tag{9.3}$$

For all these figures $s = 50$. It is important to note that every single point of the $s = 50$ points that are used to calculate the boxplot, std or SEM, is calculated by averaging over all the $n = 276$ measurements. This could be the explanation about why the curves of the implementations in this thesis are significantly smoother than the curves from [25]. In their case, $s = 100$, but $n = 6$.

Figure 9.10 illustrates two things. First, there are no obvious outliers. Second, the distances between the first quartile and the median and between the third quartile and the median are smaller than the distances between the first quartile and the minimum and between the third quartile and the maximum. This is expected because a Gaussian dis-

tribution is expected because of the central limit theorem [26]. Because of that Gaussian distribution, figure 9.11 can give valuable information. This figure shows relatively small error bars which means that *std* is small, which means that the values tend to be close to the mean. Figure 9.12 has to be made very large in order to have visible error bars. This means that standard error on the mean is very small. That was not unexpected because the curve is very smooth. Because the curves are already smooth and the standard errors are very small for every point, fitting a curve would not make a big difference. Therefore, no curve is fitted.

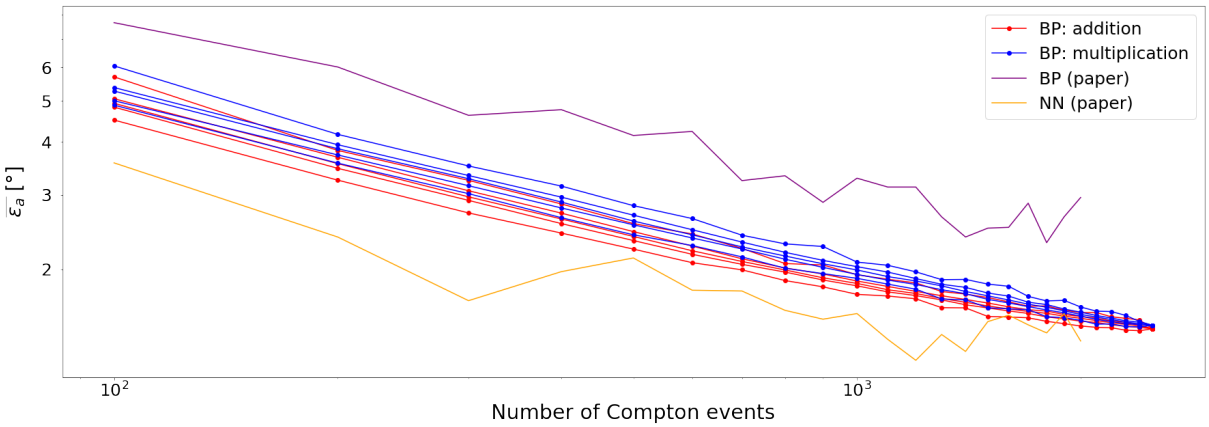


Figure 9.10: Comparison of angular errors $\overline{\varepsilon_a}$ in function of number of Compton events for the different implementations in this thesis and the implementations in [25] with the addition of simplified box plots

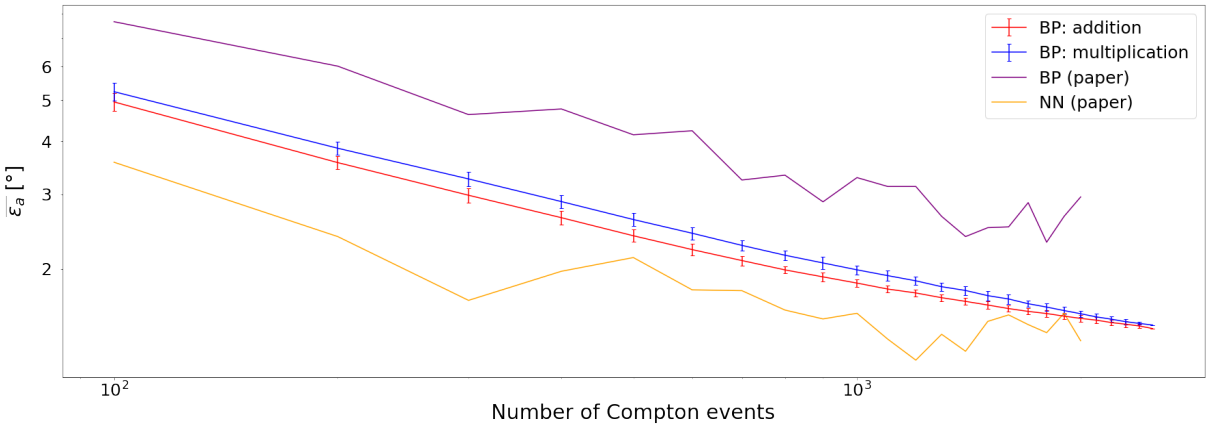


Figure 9.11: Comparison of angular errors $\overline{\varepsilon_a}$ in function of number of Compton events for the different implementations in this thesis and the implementations in [25] with the addition of error bars representing std

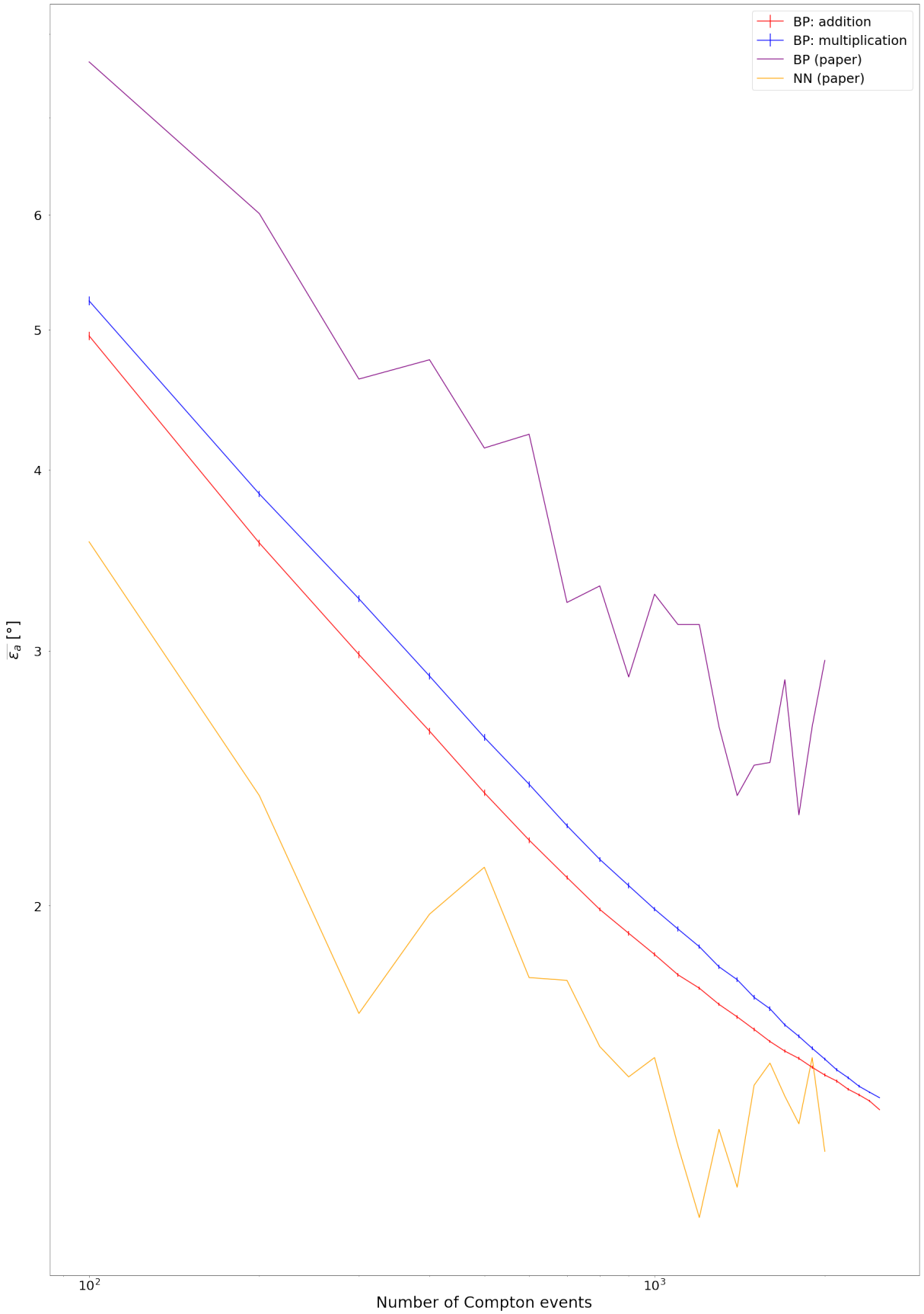


Figure 9.12: Comparison of angular errors $\overline{\varepsilon_a}$ in function of number of Compton events for the different implementations in this thesis and the implementations in [25] with the addition of error bars representing SEM

9.7 $\overline{\varepsilon_a}$ in function of measurement time

Figure 9.9 demonstrated that,using the number of Compton events as figure of merit, the neural network from [25] outperforms the back-projection algorithms implemented in this thesis. However, in [25] a very limited number of frames are considered usable frames. Only those frames with a total energy between 647 keV and 670 keV are considered usable. In this thesis $R_e = 30$ keV or $R_e = 35$ keV is mostly used. Even energy ranges as large as $R_e = 50$ keV are considered. This means that this thesis uses a larger percentage of the generated frames. Table 9.10 presents these percentages, referred to as $uf\%$. It can be noted that even though the steps are always 5 keV, the percentage of additional usable frames decreases every time. That is because the full absorption peak is bell-shaped (see fig. 9.2). Because this research uses a larger percentage of generated frames, the angular error $\overline{\varepsilon_a}$ is also calculated depending on the measuring time for a source with an activity of 1 MBq.

Table 9.10: $uf\%$ for different values of R_e in this thesis and the percentage for paper [25]

	R_e = **30 keV**	R_e = **35 keV**	R_e = **40 kev**	R_e = **45 keV**	R_e = **50 keV**	**Paper [25]**
$uf\%$ [%]	0.479	0.572	0.653	0.718	0.765	0.186

9.7.1 Implemented calculations

In order to plot $\overline{\varepsilon_a}$ in function of the measurement time, the y-values can remain the same if the x-values are transformed. This is done using the steps explained below. Figure 9.13 visualises these.

Figure 9.13: Visualisation of the steps necessary to transform the x-axis

Step 1: Calculate the usable frames per hour for a source with A = 1 MBq

With uf_h = usable frames per hour per MBq $\left[\frac{\text{frames}}{\text{h·MBq}}\right]$, A = source activity [MBq], t_m = measurement time [h], uf = number of usable frames, uf_h is calculated using equation 9.4.

$$uf_h = \frac{1}{n}\sum_{i=1}^{n}\frac{uf}{t_m \cdot A} \tag{9.4}$$

The results are presented in table 9.11.

Table 9.11: uf_h for different values of R_e in this thesis and for paper [25]

	R_e = **30 keV**	R_e = **35 keV**	R_e = **40 kev**	R_e = **45 keV**	R_e = **50 keV**	**Paper [25]**
uf_h $\left[\frac{\text{frames}}{\text{h·MBq}}\right]$	283	338	386	424	452	85

Step 2: Calculate the time necessary to gather 100 usable frames for a source with A = 1 MBq

In figure 9.9 the x-axis was defined using steps of $l_s = 100$. Therefore, the time necessary to gather 100 usable frames (assuming a source with A = 1 MBq), referred to as t_{100} [h], is calculated using equation 9.5.

$$t_{100} = \frac{100}{uf_h} \tag{9.5}$$

Table 9.12: t_{100} for different values of R_e in this thesis and for paper [25]

	R_e = **30 keV**	R_e = **35 keV**	R_e = **40 kev**	R_e = **45 keV**	R_e = **50 keV**	**Paper [25]**
t_{100} [h]	0.353	0.296	0.259	0.236	0.221	1.176

Step 3: Create the x-axis

In order to keep the same y-values, the x-values need to modelled so that the first value is for 100 cones, the second for 200 cones and the 25$^{\text{th}}$ for 2500 cones. While the x-values used for the previous figures were the same for every set of y-values, this is not the case now. For every different value of R_e, a different set of x-values has to be made. It is still a set of 25 values, but it starts at t_{100} for that specific value of R_e and ends at $25 \cdot t_{100}$ for that specific value of R_e.

9.7.2 Resulting figures

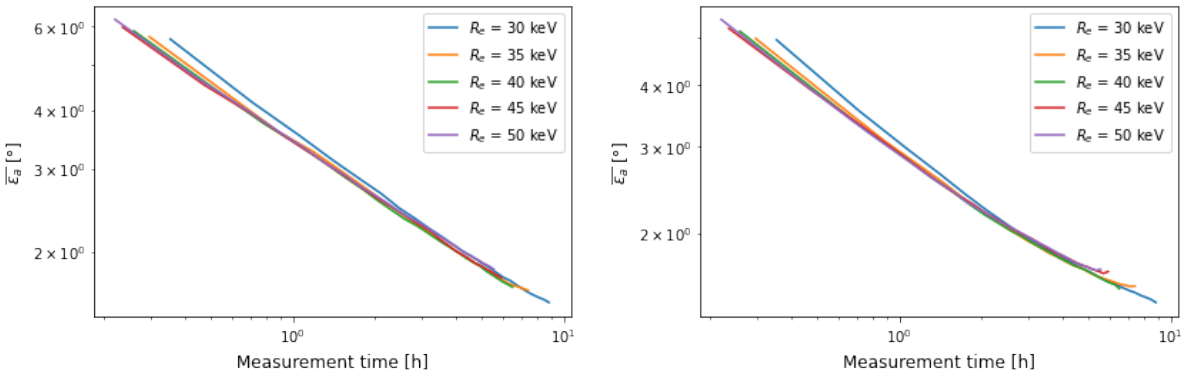
In this section different plots of $\overline{\varepsilon_a}$ in function of the measurement time will be displayed. These figures are made using the steps explained in section 9.7.1.

Addition

Figure 9.14 displays $\overline{\varepsilon_a}$ in function of the measurement time for R_e ranging from 30 keV to 50 keV and (a) $\sigma = 2°$ and (b) $\sigma = 3°$. These figures again display an exponential descent, which is logical because the adaptation of the x-axis did need distort the curves, it simply shifted them horizontally. Figure 9.15 illustrates again that $\sigma = 3°$ gives better results than $\sigma = 2°$. This is expected because for both values of σ the curves for $R_e = 30$ keV are shifted an equal distance, both curves for $R_e = 35$ keV are also shifted an equal distance (but a different distance than for $R_e = 30$ keV). This is true for all values of R_e. Therefore, the only possibility was that $\sigma = 3°$ gives better results than $\sigma = 2°$, when using the measurement time as figure of merit.

Multiplication

Figure 9.16 displays $\overline{\varepsilon_a}$ in function of the measurement time for multiplication with $\sigma = 2°$ and $v_c = 0.05$. For this figure, the same observations as those described for addition can be made.



(a) (b)

Figure 9.14: $\overline{\varepsilon_a}$ in function of the measurement time for addition for varying values of R_e and (a) $\sigma = 2°$ and (b) $\sigma = 3°$

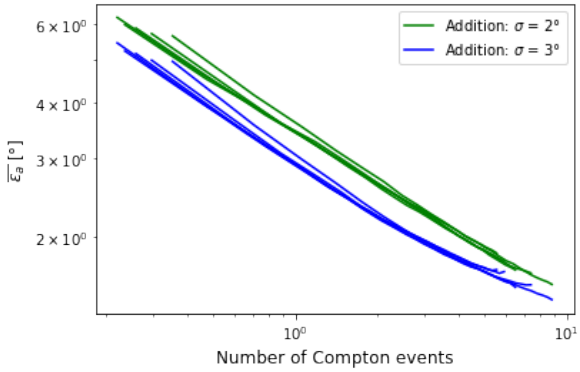


Figure 9.15: Comparison of $\overline{\varepsilon_a}$ in function of the measurement time for addition for $\sigma = 2°$ and $\sigma = 3°$

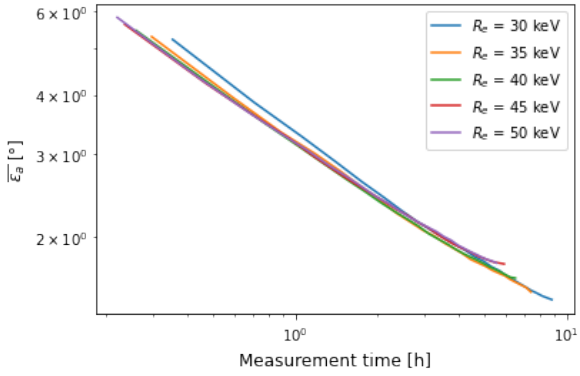


Figure 9.16: $\overline{\varepsilon_a}$ in function of the measurement time for multiplication with $\sigma = 2°$ and $v_c = 0.05$

Comparison

The previous figures illustrated that compensating for the frames used, results in very similar results for the different values of R_e. It is visible that $R_e = 30$ keV is not the best value, but apart from that value, it is difficult to determine which value for R_e gives the 'best' results. An option would be to fit a straight line for every value of R_e and use that to determine the best value. However, it is important to remember that these

curves are only estimates. Each one is made by selecting combinations of usable frames on a semi-random basis. That means that these curves would look slightly different if other random seeds were used. The difference between the curves is therefore considered insignificant. The middle value, $R_e = 40$ keV, is chosen to compare addition with $\sigma = 2°$, addition with $\sigma = 3°$ and multiplication with $\sigma = 2°$ and $v_c = 0.05$. Figure 9.17 illustrates this comparison. Because R_e has the same value for all three implementations, they all shift the same distance, which means that, again, multiplication performs better than addition with $\sigma = 2°$ but worse than addition with $\sigma = 3°$.

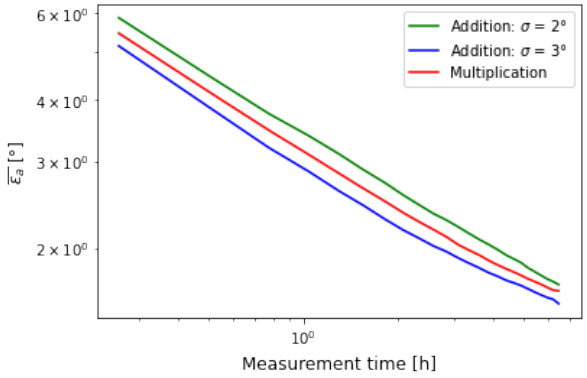


Figure 9.17: Comparison of $\overline{\varepsilon_a}$ in function of the measurement time with $R_e = 30$ keV for addition with $\sigma = 2°$, addition with $\sigma = 3°$ and multiplication with $\sigma = 2°$ and $v_c = 0.05$

9.7.3 Comparison with state-of-the-art

The implemented algorithms will again be compared to [25], this time using the measurement time as figure of merit. Figure 9.18 compares the back-projection implementation of this thesis with addition ($R_e = 30$ keV, $\sigma = 3°$), the back-projection implementation of this thesis with multiplication ($R_e = 30$ keV, $\sigma = 2°$ and $v_c = 0.05$), the back-projection implementation from [25] and the neural network from [25].

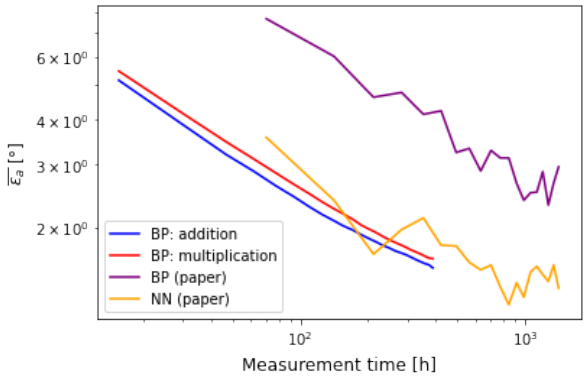


Figure 9.18: Comparison of angular errors $\overline{\varepsilon_a}$ in function of measurement time for the different implementations in this thesis and the implementations in [25]

To have a better idea about the data, the same replacements and additions are made as in the previous section. First, a simplified box plot is added. This results in figure 9.19. Second, error bars representing *std* are added. This results in figure 9.20. Third, error bars representing SEM are added. This results in figure 9.21.

Although the y-values are not exactly the same as in section 9.6.3, because section 9.6.3 used $R_e = 30$ keV and this section uses $R_e = 40$ kev, the discussion of the figures is very similar.

Figure 9.19 shows no obvious outliers and has the expected Gaussian distribution. Figure 9.20 has small error bars which means small values for *std* which indicates that the values tend to be close to the mean. Figure 9.21 displays small error bars which indicate the standard error on the mean is small. Because the curves are already smooth and the standard errors are very small for every point, no curve was fitted.

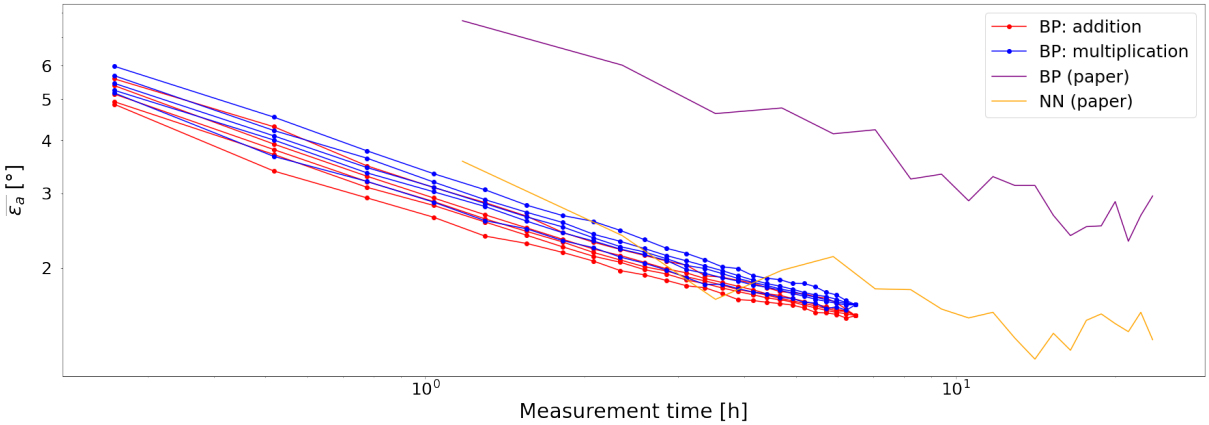


Figure 9.19: Comparison of angular errors $\overline{\varepsilon_a}$ in function of measurement time for the different implementations in this thesis and the implementations in [25] with the addition of simplified box plots

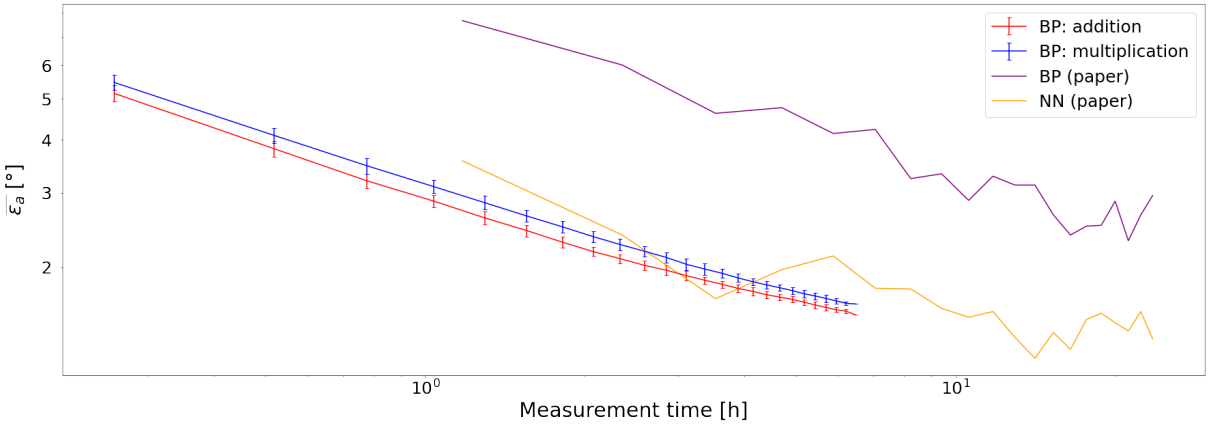


Figure 9.20: Comparison of angular errors $\overline{\varepsilon_a}$ in function of measurement time for the different implementations in this thesis and the implementations in [25] with the addition of error bars representing std

Because [25] uses only a small percentage of the generated frames and therefore needs more time to gather 100 usable frames, the curves of the back-projection and neural network implementation of [25] are not shifted the same distance as the curves of back-projection with addition and back-projection with multiplication implemented for this thesis. This results in the implemented algorithms being able to compete with the neural network from [25]. Therefore, it can be concluded that the used technology combined with the implemented algorithms can compete with the current state-of-the-art.

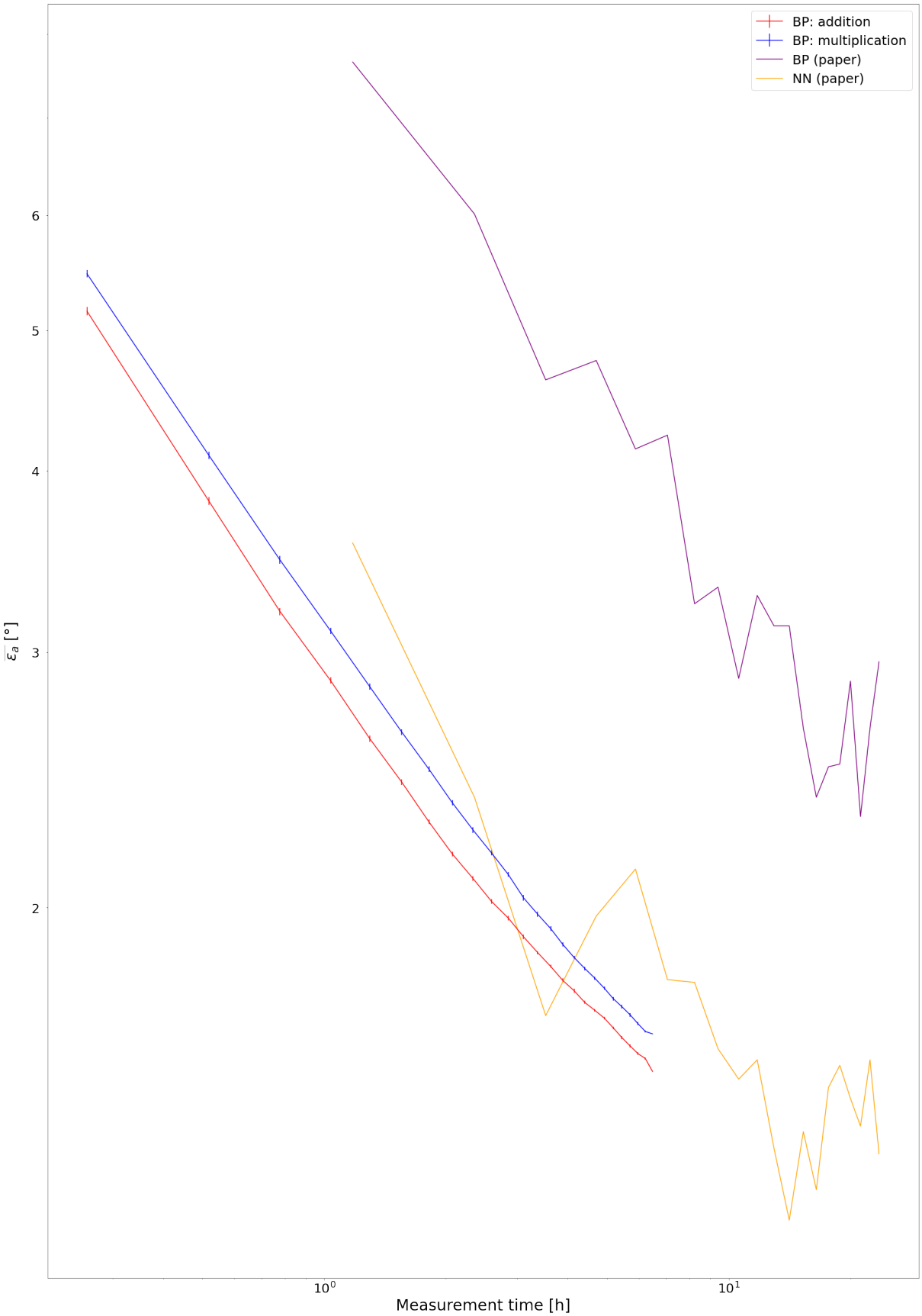


Figure 9.21: Comparison of angular errors $\overline{\varepsilon_a}$ in function of measurement time for the different implementations in this thesis and the implementations in [25] with the addition of error bars representing SEM

Chapter 10

Conclusion and outlook

This thesis researched the possibility of using a single layer Compton camera to determine the position of a γ-ray point source. Two different back-projection algorithms were implemented to accomplish that: one uses addition while the other uses multiplication. The difference in computation time is insignificant. To speed up the process, the zoom function can be used, which can make some major steps in the calculations up to 160 times faster.

The algorithms produce similar results and show an exponential descent for the angular error in function of the number of Compton interactions. Typically, addition is 6% more accurate than multiplication — but that percentage is larger for shorter measurement times and shrinks to zero for longer measurement times. Both algorithms in this thesis outperform the back-projection implementation from [25], but cannot compete with their neural network implementation. Back-projection using multiplication with $R_e = 30$ keV has an angular error of 5,2° at 100 Compton cones and an angular error of 2.0° at 1000 Compton cones, while the neural network from [25] has an angular error of 3.6° at 100 Compton cones and an angular error of 1.6° at 1000 Compton cones.

When compensating for the numbers of frames used, the figure of merit changes to the measurement time for a source with an activity of 1 MBq. Comparison between the two algorithms implemented in this thesis is the same as in the previous paragraph, but the comparison with the implementations from [25] changes: both algorithms implemented in this thesis still outperform the back-projection implementation from [25], but can now also compete with their neural network. Back-projection using addition with $R_e = 40$ keV needs 55 minutes to reach an angular error of 3° and 161 minutes to reach an angular error of 2°. The neural network needs 95 minutes to reach an angular error of 3° and 173 minutes to reach an angular error of 2°. These numbers illustrate that the results from this thesis can compete with the current state-of-the-art.

Because the γ camera combined with the implemented algorithms can compete with the current state-of-the-art, this research can be used as a first step in the localisation of a γ ray point source. However, this research involved a simplified experimental setup in a controlled environment, where the distance from the point source to the detector was known. Therefore, further research and development is necessary in order to use the single layer Compton camera in a real-life nuclear decommissioning setting.

There are several promising next steps for further research and development. The first would be analysing a second data set. The measurements for such a data set are already performed, but the data was not available on time to be analysed as part of this thesis. However, a paper will be written that includes this second data set. A link to this paper will be published on the GitHub page of this thesis [23]. Some other options for improvement could require only small adjustments, like adding compensation for detector temperature in the calculations, while others might consist of completely new implementations. A machine learning algorithm can possibly be trained to select the correct cone before doing back-projection; or a neural network can be implemented in an attempt to achieve a predetermined accuracy with less measurement time. These options all target more accurate or faster results but start from the same limited measurement setup.

To use new research in a real-life setting, it will be very important to look beyond that simplified experimental setup in a controlled environment. What happens if testing is done with a different source? What if the source is not a point source? What if there are multiple sources? In order to answer these questions, at least two things need to be done: first, the current algorithms need to be optimised and new algorithms need to be implemented. Second, it might prove very helpful to combine different types of equipment. Adding the imaging from a 3D camera, for example, might help narrow down positions of point sources.

Overall then, the technology analysed here combined with the implemented algorithms can form a solid point of departure, with more research needed to unlock its full potential.

Bibliography

[1] R. Volk, F. Hübner, T. Hünlich, and F. Schultmann, "The future of nuclear decommissioning – A worldwide market potential study," *Energy Policy*, vol. 124, pp. 226–261, Jan. 2019. DOI: 10.1016/j.enpol.2018.08.014.

[2] CDC, *ALARA - As Low As Reasonably Achievable*, May 2022. [Online]. Available: https://www.cdc.gov/nceh/radiation/alara.html (visited on 11/06/2022).

[3] M. Simons, "Localisation of hot spots during nuclear decommissioning with a CdTe gamma camera," M.S. thesis, UHasselt and KULeuven, Diepenbeek, Jun. 2020.

[4] D. Turecek, J. Jakubek, E. Trojanova, and L. Sefc, "Compton camera based on Timepix3 technology," *Journal of Instrumentation*, vol. 13, no. 11, pp. C11022–C11022, Nov. 2018. DOI: 10.1088/1748-0221/13/11/C11022.

[5] D. Turecek, J. Jakubek, E. Trojanova, and L. Sefc, "Single layer Compton camera based on Timepix3 technology," *Journal of Instrumentation*, vol. 15, no. 01, pp. C01014–C01014, Jan. 2020. DOI: 10.1088/1748-0221/15/01/C01014.

[6] M. F. L'Annunziata, *Radioactivity: introduction and history*. Oxford: Elsevier, 2007, ISBN: 978-1-281-11984-1.

[7] CDC, *Penetration Abilities of Different Types of Radiation*. [Online]. Available: https://www.cdc.gov/training/products/RN/page4976.html (visited on 05/01/2023).

[8] G. Gilmore, *Practical Gamma-ray Spectrometry*, 2nd ed. Warrington: John Wiley & Sons Ltd, 2008.

[9] G. F. Knoll, *Radiation detection and measurement*, 4th ed. Hoboken, N.J: John Wiley, 2010, ISBN: 978-0-470-13148-0.

[10] ADVACAM, *AdvaPIX TPX3*. Nov. 2020. [Online]. Available: https://advacam.com/wp-content/uploads/2022/03/APXT3M-Xxx201030-AdvaPIX-TPX3-Datasheet-2020-11-11.pdf (visited on 12/09/2022).

[11] T. Poikela, J. Plosila, T. Westerlund, *et al.*, "Timepix3: A 65K channel hybrid pixel readout chip with simultaneous ToA/ToT and sparse readout," *Journal of Instrumentation*, vol. 9, no. 05, p. C05013, May 2014. DOI: 10.1088/1748-0221/9/05/C05013.

[12] B. Bergmann, M. Pichotka, S. Pospisil, *et al.*, "3D track reconstruction capability of a silicon hybrid active pixel detector," *The European Physical Journal C*, vol. 77, no. 6, p. 421, Jun. 2017. DOI: 10.1140/epjc/s10052-017-4993-4.

[13] Palomar Technologies, *Solder Bump Bonding, Ball Bumps and Wire Bonds*. [Online]. Available: https://www.palomartechnologies.com/blog/bid/28476/Solder-Bump-Bonding-Ball-Bumps-and-Wire-Bonds (visited on 02/20/2023).

[14] M. Campbell, E. Heijne, T. Holý, *et al.*, "Study of the charge sharing in a silicon pixel detector by means of α-particles interacting with a Medipix2 device," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 591, no. 1, pp. 38–41, Jun. 2008. DOI: 10.1016/j.nima.2008.03.096.

[15] L. Rossi, *Pixel detectors: from fundamentals to applications*. Berlin, New York: Springer, 2006, ISBN: 978-3-540-28332-4.

[16] J. Jakubek, A. Cejnarova, M. Platkevic, J. Solc, and Z. Vykydal, "Event by event energy sensitive imaging with TimePix pixel detector and its application for gamma photon tracking," in *2008 IEEE Nuclear Science Symposium Conference Record*, Dresden, Germany: IEEE, 2008, pp. 3451–3458. DOI: 10.1109/NSSMIC.2008.4775081.

[17] D. Turecek, J. Jakubek, and P. Soukup, "USB 3.0 readout and time-walk correction method for Timepix3 detector," *Journal of Instrumentation*, vol. 11, no. 12, pp. C12065–C12065, Dec. 2016. DOI: 10.1088/1748-0221/11/12/C12065.

[18] J. Jakubek, "Precise energy calibration of pixel detector working in time-over-threshold mode," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 633, S262–S266, May 2011. DOI: 10.1016/j.nima.2010.06.183.

[19] T. Billoud, C. Leroy, C. Papadatos, M. Pichotka, S. Pospisil, and J. Roux, "Characterization of a pixelated CdTe Timepix detector operated in ToT mode," *Journal of Instrumentation*, vol. 12, no. 01, P01018–P01018, Jan. 2017. DOI: 10.1088/1748-0221/12/01/P01018.

[20] B. Bergmann, P. Burian, P. Manek, and S. Pospisil, "3D reconstruction of particle tracks in a 2 mm thick CdTe hybrid pixel detector," *The European Physical Journal C*, vol. 79, no. 2, p. 165, Feb. 2019. DOI: 10.1140/epjc/s10052-019-6673-z.

[21] ADVACAM, *PiXet Pro*, 2020. [Online]. Available: https://downloads.advacam.com/index.php (visited on 12/09/2022).

[22] Python Software Foundation, *Python 3.9*. [Online]. Available: https://www.python.org (visited on 12/09/2022).

[23] A. Michiels, *Position-determination-gamma-source-Compton-camera*. [Online]. Available: https://github.com/anoukmichiels/position-determination-gamma-source-Compton-camera (visited on 06/10/2023).

[24] T. Schoonjans, A. Brunetti, B. Golosio, *et al.*, "The xraylib library for X-ray–matter interactions. Recent developments," *Spectrochimica Acta Part B: Atomic Spectroscopy*, vol. 66, no. 11, pp. 776–784, Nov. 2011. DOI: 10.1016/j.sab.2011.09.011.

[25] G. Daniel, Y. Gutierrez, and O. Limousin, "Application of a deep learning algorithm to Compton imaging of radioactive point sources with a single planar CdTe pixelated detector," *Nuclear Engineering and Technology*, vol. 54, no. 5, pp. 1747–1753, May 2022. DOI: 10.1016/j.net.2021.10.031.

[26] C. Kamis and S. M.Lynch, *Central Limit Theorem*. London: SAGE Publications Ltd, 2020, ISBN: 978-1-5297-5009-6.

[27] A. Rohatgi, *Webplotdigitizer: Version 4.6*, 2022. [Online]. Available: https://automeris.io/WebPlotDigitizer (visited on 06/10/2023).

Appendix A

Measurements overview

This appendix gives an overview of all the measurements. **Phi** and **Theta** are the measured phi and theta for that measurement. The **Time** is the measurement time in hours. **Counts** is the total number of frames. **uf (30)** is the number of usable frames for $R_e =$ 30 keV. **Cones (30)** is the number of cones for $R_e = 30$ keV. **uf (35)** is the number of usable frames for $R_e = 35$ keV. **Cones (35)** is the number of cones for $R_e = 35$ keV.

Table A.1: Overview of measurements

Index	Phi	Theta	Time [h]	Counts	uf (30)	Cones (30)	uf (35)	Cones (35)
0	-54.9	-54.9	1	593192	3803	6620	4585	8024
1	-54.9	-49.95	1	622254	3963	6922	4720	8275
2	-54.9	-44.55	1	651899	4042	7118	4805	8459
3	-54.9	-39.6	1	677702	4048	7143	4874	8623
4	-54.9	-34.65	1	696783	4179	7390	4953	8796
5	-54.9	-29.7	1	717790	4031	7155	4856	8638
6	-54.9	-24.75	1	735257	4080	7220	4852	8613
7	-54.9	-19.8	1	750501	4124	7274	4925	8714
8	-54.9	-14.85	1	764769	4241	7507	5057	8979
9	-54.9	-9.9	1	769896	4104	7351	4968	8909
10	-54.9	-4.95	1	772334	4130	7381	4951	8845
11	-54.9	-0.0	1	776112	4059	7213	4948	8824
12	-54.9	5.4	1	777867	4016	7208	4873	8732
13	-54.9	10.35	1	780102	3949	6993	4745	8428
14	-54.9	15.3	1	772820	3666	6471	4507	7980
15	-54.9	20.25	1	766811	2984	5302	3797	6780
16	-54.9	25.2	1	752319	3026	5354	3893	6902
17	-54.9	30.15	1	734175	3161	5618	3996	7116
18	-54.9	35.1	1	713697	3403	6010	4300	7613
19	-54.9	40.05	1	690279	3760	6617	4622	8144
20	-54.9	45.0	1	663251	3864	6773	4742	8337
21	-54.9	50.4	1	629513	3782	6593	4690	8208
22	-54.9	54.9	1	600754	4011	6973	4869	8504
23	-50.4	-54.9	1	618735	3769	6605	4534	7971
24	-50.4	-50.4	1	645099	3913	6806	4725	8254
25	-50.4	-45.0	1	676687	4011	7060	4795	8456
26	-50.4	-40.05	1	703275	3836	6790	4611	8174
27	-50.4	-35.1	1	728591	4002	7107	4838	8613

Continued on next page

Table A.1 – continued from previous page

Index	Phi	Theta	Time [h]	Counts	uf (30)	Cones (30)	uf (35)	Cones (35)
28	-50.4	-30.15	1	748876	3770	6776	4579	8224
29	-50.4	-25.2	1	770236	3395	6019	4191	7440
30	-50.4	-20.25	1	788749	3103	5505	3933	7018
31	-50.4	-15.3	1	803322	3211	5735	4024	7219
32	-50.4	-10.35	1	810223	3928	7021	4831	8651
33	-50.4	-5.4	1	812943	4043	7201	4857	8686
34	-50.4	-0.0	1	816206	3973	7146	4785	8624
35	-50.4	4.95	1	819771	4107	7397	4961	8950
36	-50.4	9.9	1	821196	4257	7642	5132	9216
37	-50.4	14.85	1	816497	4087	7351	4880	8797
38	-50.4	19.8	1	807049	4225	7498	5066	9032
39	-50.4	24.75	1	794309	4284	7631	5127	9156
40	-50.4	29.7	1	778310	4273	7609	5142	9169
41	-50.4	34.65	1	757335	4098	7283	4967	8843
42	-50.4	39.6	1	733048	4109	7253	4982	8833
43	-50.4	44.55	1	703915	4124	7257	4966	8770
44	-50.4	49.95	1	668565	4116	7215	4973	8741
45	-50.4	54.9	1	631129	4070	7105	4920	8615
46	-45.0	-54.9	1	647014	3921	6875	4723	8305
47	-45.0	-49.95	1	683294	4037	7109	4816	8489
48	-45.0	-44.55	1	715669	3937	6991	4714	8372
49	-45.0	-39.6	1	742216	4115	7288	4896	8689
50	-45.0	-34.65	1	769036	4097	7288	4937	8799
51	-45.0	-29.7	1	794875	4184	7487	5003	8973
52	-45.0	-24.75	1	818258	4150	7464	4984	8964
53	-45.0	-19.8	1	833926	4128	7504	4944	8994
54	-45.0	-14.85	1	848368	4215	7619	5041	9140
55	-45.0	-9.9	1	856900	4271	7722	5043	9128
56	-45.0	-4.95	1	859809	4063	7345	4890	8856
57	-45.0	-0.0	1	863165	4203	7617	5024	9122
58	-45.0	5.4	1	867510	4113	7458	4914	8905
59	-45.0	10.35	1	866670	3991	7278	4819	8793
60	-45.0	15.3	1	862284	3936	7104	4810	8690
61	-45.0	20.25	1	854279	3897	7073	4713	8569
62	-45.0	25.2	1	836133	4012	7188	4864	8728
63	-45.0	30.15	1	817867	3858	6902	4733	8488
64	-45.0	35.1	1	794434	3771	6714	4665	8328
65	-45.0	40.05	1	767609	4145	7412	4994	8927
66	-45.0	45.0	1	737095	4182	7397	5077	8981
67	-45.0	50.4	1	698809	4021	7120	4822	8558
68	-45.0	54.9	1	666861	4020	7042	4837	8502
69	-40.05	-54.9	1	673891	3952	6989	4769	8465
70	-40.05	-50.4	1	702161	4078	7164	4929	8680
71	-40.05	-45.0	1	736590	4100	7307	4884	8711
72	-40.05	-40.05	1	767969	4078	7342	4850	8750
73	-40.05	-35.1	1	798557	4147	7452	4955	8916
74	-40.05	-30.15	1	824497	4030	7261	4833	8713
75	-40.05	-25.2	1	849100	4004	7242	4832	8751
76	-40.05	-20.25	1	868476	3981	7203	4779	8666
77	-40.05	-15.3	1	881528	3841	6980	4668	8505
78	-40.05	-10.35	1	894895	3942	7203	4750	8688
79	-40.05	-5.4	1	898525	4104	7475	4903	8937

Continued on next page

Table A.1 – continued from previous page

Index	Phi	Theta	Time [h]	Counts	uf (30)	Cones (30)	uf (35)	Cones (35)
80	-40.05	-0.0	1	901179	4141	7556	4950	9042
81	-40.05	4.95	1	907169	4193	7657	5047	9227
82	-40.05	9.9	1	906510	4105	7507	4942	9037
83	-40.05	14.85	1	902900	4221	7696	5062	9250
84	-40.05	19.8	1	893562	4202	7656	5048	9214
85	-40.05	24.75	1	876215	4260	7686	5126	9285
86	-40.05	29.7	1	856002	4371	7879	5178	9334
87	-40.05	34.65	1	837318	4367	7860	5196	9380
88	-40.05	39.6	1	808132	4242	7591	5055	9068
89	-40.05	44.55	1	777751	4399	7796	5213	9270
90	-40.05	49.95	1	738629	4368	7768	5188	9213
91	-40.05	54.9	1	907771	4323	7656	5185	9187
92	-35.1	-54.9	1	694969	4475	7911	5224	9265
93	-35.1	-49.95	1	733101	4593	8131	5309	9394
94	-35.1	-44.55	1	768561	4684	8365	5432	9718
95	-35.1	-39.6	1	799154	4572	8218	5358	9622
96	-35.1	-34.65	1	833449	4557	8192	5282	9489
97	-35.1	-29.7	1	862496	4501	8181	5253	9572
98	-35.1	-24.75	1	887098	4503	8239	5226	9571
99	-35.1	-19.8	1	912794	4593	8423	5294	9720
100	-35.1	-14.85	1	926960	4566	8380	5325	9765
101	-35.1	-9.9	1	929200	4434	8149	5119	9411
102	-35.1	-4.95	1	934112	4442	8163	5200	9542
103	-35.1	-0.0	1	942321	4202	7748	4974	9165
104	-35.1	5.4	1	949020	4245	7817	5079	9365
105	-35.1	10.35	1	943390	4082	7491	4893	8994
106	-35.1	15.3	1	938081	4130	7600	4905	9022
107	-35.1	20.25	1	925419	4226	7749	5008	9195
108	-35.1	25.2	1	910667	4261	7742	5061	9212
109	-35.1	30.15	1	887413	4294	7778	5137	9329
110	-35.1	35.1	1	861341	4414	7957	5277	9537
111	-35.1	40.05	1	832502	4396	7937	5204	9408
112	-35.1	45.0	1	813072	4461	7951	5260	9385
113	-35.1	50.4	1	757783	4355	7710	5173	9155
114	-35.1	54.9	1	725369	4267	7548	5127	9062
115	-30.15	-54.9	1	711026	4568	8080	5328	9443
116	-30.15	-50.4	1	743391	4740	8394	5500	9736
117	-30.15	-45.0	1	782873	4622	8287	5354	9607
118	-30.15	-40.05	1	816741	4722	8451	5401	9666
119	-30.15	-35.1	1	851654	4664	8447	5354	9693
120	-30.15	-30.15	1	880770	4610	8416	5348	9764
121	-30.15	-25.2	1	905772	4503	8281	5210	9571
122	-30.15	-20.25	1	929081	4622	8462	5358	9839
123	-30.15	-15.3	1	946902	4526	8416	5245	9745
124	-30.15	-10.35	1	959096	4402	8167	5137	9545
125	-30.15	-5.4	1	962428	4456	8257	5242	9714
126	-30.15	-0.0	1	968855	4418	8202	5128	9529
127	-30.15	4.95	1	974094	4563	8403	5244	9689
128	-30.15	9.9	1	975142	4566	8455	5264	9761
129	-30.15	14.85	1	971111	4697	8672	5435	10044
130	-30.15	19.8	1	963225	4671	8578	5378	9886
131	-30.15	24.75	1	941534	4762	8750	5496	10109

Continued on next page

Table A.1 – continued from previous page

Index	Phi	Theta	Time [h]	Counts	uf (30)	Cones (30)	uf (35)	Cones (35)
132	-30.15	29.7	1	919455	4931	8996	5648	10306
133	-30.15	34.65	1	894799	4773	8679	5556	10108
134	-30.15	39.6	1	866426	4868	8784	5620	10132
135	-30.15	44.55	1	833952	4912	8771	5668	10129
136	-30.15	49.95	1	787177	4798	8529	5518	9805
137	-30.15	54.9	1	741405	4856	8610	5611	9953
138	-25.2	-54.9	1	726176	3989	7071	4828	8573
139	-25.2	-49.95	1	770838	3998	7111	4809	8580
140	-25.2	-44.55	1	807757	4165	7492	4932	8886
141	-25.2	-39.6	1	843818	3963	7173	4744	8604
142	-25.2	-34.65	1	880782	4016	7363	4832	8878
143	-25.2	-29.7	1	911954	3923	7158	4686	8575
144	-25.2	-24.75	1	936085	3773	6992	4615	8554
145	-25.2	-19.8	1	956246	3868	7115	4699	8677
146	-25.2	-14.85	1	973643	3864	7131	4640	8569
147	-25.2	-9.9	1	984806	4068	7499	4871	9003
148	-25.2	-4.95	1	991917	4133	7652	4933	9156
149	-25.2	-0.0	1	993325	3906	7259	4657	8656
150	-25.2	5.4	1	1000485	4192	7753	4970	9208
151	-25.2	10.35	1	1002616	4254	7898	5022	9312
152	-25.2	15.3	1	993097	4240	7839	5020	9285
153	-25.2	20.25	1	981583	4542	8344	5292	9740
154	-25.2	25.2	1	964026	4621	8469	5365	9837
155	-25.2	30.15	1	938645	4664	8568	5434	9990
156	-25.2	35.1	1	912709	4905	8907	5669	10312
157	-25.2	40.05	1	881308	4849	8737	5566	10031
158	-25.2	45.0	1	845617	4912	8819	5659	10146
159	-25.2	50.4	1	799423	4902	8747	5659	10112
160	-25.2	54.9	1	761428	4823	8511	5615	9928
161	-20.25	-54.9	1	740770	4002	7107	4776	8492
162	-20.25	-50.4	1	772684	4049	7224	4849	8664
163	-20.25	-45.0	1	812642	4156	7475	4953	8932
164	-20.25	-40.05	1	852902	4170	7556	4966	9014
165	-20.25	-35.1	1	892866	3960	7256	4746	8706
166	-20.25	-30.15	1	921945	4067	7463	4773	8770
167	-20.25	-25.2	1	947937	3889	7160	4689	8661
168	-20.25	-20.25	1	972974	3782	6986	4553	8405
169	-20.25	-15.3	1	992593	3822	7105	4598	8549
170	-20.25	-10.35	1	1004582	3808	7056	4549	8426
171	-20.25	-5.4	1	1010211	3781	7044	4604	8593
172	-20.25	-0.0	1	1016951	3598	6647	4386	8107
173	-20.25	4.95	1	1021951	3760	7001	4505	8376
174	-20.25	9.9	1	1020825	3847	7152	4648	8638
175	-20.25	14.85	1	1017345	3828	7085	4641	8608
176	-20.25	19.8	1	1005575	3967	7327	4735	8765
177	-20.25	24.75	1	986721	3899	7154	4724	8683
178	-20.25	29.7	1	964568	3996	7307	4844	8858
179	-20.25	34.65	1	938696	4167	7618	5035	9223
180	-20.25	39.6	1	906713	4315	7839	5151	9357
181	-20.25	44.55	1	868714	4319	7853	5143	9343
182	-20.25	49.95	1	827209	4480	7970	5324	9506
183	-20.25	54.9	1	776061	4418	7836	5293	9406

Continued on next page

Table A.1 – continued from previous page

Index	Phi	Theta	Time [h]	Counts	uf (30)	Cones (30)	uf (35)	Cones (35)
184	-15.3	-54.9	1	747843	3997	7086	4754	8454
185	-15.3	-49.95	1	789807	4064	7268	4870	8724
186	-15.3	-44.55	1	834914	4035	7293	4895	8865
187	-15.3	-39.6	1	872231	3877	7051	4703	8567
188	-15.3	-34.65	1	911475	3744	6879	4550	8362
189	-15.3	-29.7	1	942422	3765	6964	4556	8431
190	-15.3	-24.75	1	970970	3745	6957	4513	8393
191	-15.3	-19.8	1	990898	3591	6658	4444	8243
192	-15.3	-14.85	1	1012224	3470	6456	4229	7873
193	-15.3	-9.9	1	1024150	3367	6226	4132	7642
194	-15.3	-4.95	1	1029561	3282	6069	4028	7453
195	-15.3	-0.0	1	1033855	3408	6362	4307	8030
196	-15.3	5.4	1	1039349	3840	7144	4631	8630
197	-15.3	10.35	1	1038682	3901	7255	4648	8651
198	-15.3	15.3	1	1030503	4093	7563	4915	9083
199	-15.3	20.25	1	1018379	4196	7736	5025	9274
200	-15.3	25.2	1	1000703	4372	8050	5121	9445
201	-15.3	30.15	1	975375	4423	8149	5228	9653
202	-15.3	35.1	1	946017	4380	8014	5216	9544
203	-15.3	40.05	1	914722	4328	7896	5145	9380
204	-15.3	45.0	1	876782	4358	7828	5136	9272
205	-15.3	50.4	1	828871	4428	7887	5247	9364
206	-15.3	54.9	1	786795	4404	7800	5197	9231
207	-10.35	-54.9	1	753024	4117	7329	4931	8790
208	-10.35	-50.4	1	787224	4037	7226	4826	8660
209	-10.35	-45.0	1	830912	3923	7088	4686	8481
210	-10.35	-40.05	1	871861	3983	7241	4811	8769
211	-10.35	-35.1	1	910188	3897	7188	4679	8632
212	-10.35	-30.15	1	943601	3876	7142	4743	8759
213	-10.35	-25.2	1	971804	3825	7109	4625	8592
214	-10.35	-20.25	1	997652	3850	7120	4628	8567
215	-10.35	-15.3	1	1017252	3796	7062	4573	8511
216	-10.35	-10.35	1	1031852	3694	6880	4477	8340
217	-10.35	-5.4	1	1036908	3424	6385	4204	7841
218	-10.35	-0.0	1	1042275	3029	5609	3761	6953
219	-10.35	4.95	1	1047873	3077	5701	3895	7234
220	-10.35	9.9	1	1050457	3023	5588	3789	7022
221	-10.35	14.85	1	1043044	3082	5696	3913	7250
222	-10.35	19.8	1	1032586	3041	5627	3838	7104
223	-10.35	24.75	1	1014498	3207	5878	4008	7382
224	-10.35	29.7	1	989897	3665	6743	4467	8253
225	-10.35	34.65	1	962422	3926	7167	4781	8744
226	-10.35	39.6	1	929821	4089	7455	4957	9047
227	-10.35	44.55	1	891957	4002	7261	4847	8804
228	-10.35	49.95	1	846541	4125	7387	4981	8923
229	-10.35	54.9	1	794812	4376	7781	5213	9270
230	-5.4	-54.9	1	755369	4339	7720	5080	9043
231	-5.4	-49.95	1	798510	4401	7922	5145	9276
232	-5.4	-44.55	1	843789	4266	7743	5003	9083
233	-5.4	-39.6	1	884288	4216	7665	4956	9032
234	-5.4	-34.65	1	921692	3953	7279	4684	8631
235	-5.4	-29.7	1	953699	3835	7093	4628	8570

Continued on next page

Table A.1 – continued from previous page

Index	Phi	Theta	Time [h]	Counts	uf (30)	Cones (30)	uf (35)	Cones (35)
236	-5.4	-24.75	1	982601	3819	7063	4641	8607
237	-5.4	-19.8	1	1007349	3480	6424	4281	7912
238	-5.4	-14.85	1	1026207	3413	6353	4167	7770
239	-5.4	-9.9	1	1038169	3330	6167	4171	7736
240	-5.4	-4.95	1	1045256	3394	6320	4163	7760
241	-5.4	-0.0	1	1049507	3703	6902	4497	8379
242	-5.4	5.4	1	1054854	4063	7583	4841	9018
243	-5.4	10.35	1	1053373	4280	7947	5018	9330
244	-5.4	15.3	1	1045543	4280	7987	5003	9340
245	-5.4	20.25	1	1033308	4360	8112	5115	9513
246	-5.4	25.2	1	1015496	4331	8041	5105	9498
247	-5.4	30.15	1	989035	4335	8021	5084	9410
248	-5.4	35.1	1	961936	4424	8183	5152	9517
249	-5.4	40.05	1	925646	4478	8148	5264	9599
250	-5.4	45.0	1	889370	4558	8246	5380	9743
251	-5.4	50.4	1	841520	4379	7874	5252	9447
252	-5.4	54.9	1	798691	4383	7822	5247	9364
253	0.0	-54.9	1	755688	4339	7760	5072	9076
254	0.0	-50.4	1	790270	4426	7943	5171	9292
255	0.0	-45.0	1	836818	4404	8000	5176	9400
256	0.0	-40.05	1	875259	4422	8066	5185	9455
257	0.0	-35.1	1	916981	4367	7988	5091	9312
258	0.0	-30.15	1	948522	4364	8036	5122	9452
259	0.0	-25.2	1	976874	4371	8125	5159	9605
260	0.0	-20.25	1	1003786	4371	8129	5091	9471
261	0.0	-15.3	1	1023520	4498	8412	5203	9719
262	0.0	-10.35	1	1035411	4424	8262	5133	9596
263	0.0	-5.4	1	1043777	4384	8188	5063	9448
264	0.0	-0.0	1	1049380	4297	8025	5000	9339
265	0.0	4.95	1	1056578	3804	7081	4580	8535
266	0.0	9.9	1	1055086	3305	6161	4130	7721
267	0.0	14.85	1	1052020	3110	5771	3900	7246
268	0.0	19.8	1	1039303	3279	6087	4083	7585
269	0.0	24.75	1	1022247	2814	5184	3578	6611
270	0.0	29.7	1	999951	3033	5538	3803	6964
271	0.0	34.65	1	973209	2684	4915	3468	6366
272	0.0	39.6	1	940211	3605	6560	4467	8156
273	0.0	44.55	1	898048	4360	7895	5171	9354
274	0.0	49.95	1	853668	4329	7779	5159	9284
275	0.0	54.9	1	802174	4416	7853	5251	9360

Appendix B

Data from state-of-the-art paper [25]

This appendix gives an overview of the data extracted from [25]. The data from table B.1 is extracted from the plot in [25] using [27] and table B.2 is copied from [25].

Table B.1: x-input and y-input from the plot in [25] extracted using [27]

	x-axis	y-axis angle [degrees]	
Index	**Cones**	**Back-projection**	**Neural network**
0	100	7.65260	3.56763
1	200	6.01239	2.38310
2	300	4.62127	1.6852590
3	400	4.765354	1.9735043
4	500	4.141358	2.126273
5	600	4.233179	1.78413
6	700	3.239486	1.7763306
7	800	3.325866	1.5988566
8	900	2.8777139	1.523551
9	1000	3.2823922	1.57105
10	1100	3.12779	1.3653314
11	1200	3.127793	1.21818
12	1300	2.659266	1.40173
13	1400	2.3831032	1.27839
14	1500	2.500893	1.5036
15	1600	2.5118864	1.557331
16	1700	2.8651202	1.4774
17	1800	2.311050	1.414087
18	1900	2.659266	1.5710
19	2000	2.95444	1.353407

Table B.2: Characteristics of the measurements from [25], copied for comparison

Position of the source (longitude and latitude)	**Source activity**	**Measurement time**	**Total number of counts**	**Number of selected Compton events**
(0°,0°)	1.7 MBq	35.6 h	3190418	4855
(10°,0°)	1.7 MBq	57.7 h	5423829	8065
(30°,0°)	1.7 MBq	83.8 h	6993265	12222
(35°,0°)	1.7 MBq	19.5 h	1672290	3014
(40°,0°)	1.7 MBq	76.8 h	4791771	11581
(-22°,33°)	3.6 MBq	61.1 h	8213658	17978