Improving clustering methods for the FoCal detector

Tom Bannink [‡]

Supervisor: Dr. Ir. M. van Leeuwen
Co-supervisor: Drs. D. Lodato
Institute for Subatomic Physics
Utrecht University

June 19, 2013

Abstract

The Forward Calorimeter, FoCal for short, is a proposed detector for the ALICE project at CERN. It is an electromagnetic calorimeter with high position granularity layers allowing separation of nearby particles like the decay photons of the neutral pion. Monte Carlo simulations were used to simulate the detection of these particles and a clustering algorithm is used to reconstruct the particle locations and energies based on the detector's response. The methods used in the clustering algorithm that deal with separation of particles that are close together will be discussed in this thesis. Modifications of the algorithm are introduced which slightly improve the efficiency from 85.8% to 87.4%.

[‡]E-mail: t.r.bannink@students.uu.nl

Contents

Introduction

In order to fully understand the laws of nature, it is of paramount importance to understand the structure of the constituents of matter such as protons and neutrons. Scattering experiments are done to reveal the structure of the proton on a deep level. In order to study the inside of the proton, collisions at very high energies are needed. The Large Hadron Collider (LHC) at CERN in Switzerland can accelerate particles up to very high speeds, allowing these high energy collisions. ALICE, short for A Large Ion Collider Experiment, is one of the large experiments at the LHC. The Forward Calorimeter, FoCal, is a proposed particle detector for ALICE.

Parton is the general term for the constituents of hadrons (compound particles like the proton). These partons can be quarks, anti-quarks or gluons. Deep Inelastic Scattering experiments show that a proton consists of three quarks, called valence quarks, which are bound within the proton via the strong force carried by the gluons. Single partons are the actual particles colliding at the LHC and knowing their energy before the collision is crucial to understanding the structure of the proton. When the energy of the protons is increased in a scattering experiment, gluons will have an increased probability of colliding and these gluons will carry a low fraction of the momentum of the proton. A variable called the Bjorken-x is defined as the momentum of a parton as a fraction of the momentum of the hadron. The low x regime is the region of interest for the FoCal project.

It is predicted that the high gluon densities, typical of the low x regime, will lead to *gluon saturation* and this can be explained by a model for a new state of matter called the *Colour Glass Condensate*. As a result of parton interactions, *direct photons* are emitted and studying these photons can give insight into the gluon field at such low x values.

The FoCal detector is supposed to detect these, and the photons that are the most interesting for investigating the colour glass condensate are the ones emitted at a very small angle with the beam pipe. The detector will therefore be placed close to the beam pipe, at a large distance to the interaction point. The scattering will also produce other particles, of which one is the neutral pion. This π^0 particle will decay into two photons and these will also be measured by the detector. At high energies these two photons will have a small opening angle. The FoCal detector is a specialized calorimeter with high granularity layers that should be able to separate these nearby photons. This is necessary because only the *direct photons* are the particles of interest so the detector should be able to recognize π^0 decay products and subtract them as background signal.

An important part of this process is the software that processes the measured data. Based on the response of the detector, the software should decide where the particles hit the detector and what their energies were. This can be a difficult task if the particles hit the detector very close together, as the regions in which they deposit energy overlap.

The goal of this research is to optimize the part of the clustering algorithm that deals with separation of nearby gamma particles. This thesis will explain the steps taken and the results that were obtained.

1 Particle detection

1.1 Particle detection in general

The particles of interest for the FoCal project are electrons, positrons and photons. Each of these particles can interact with matter in several ways and this section will discuss the processes that are the most important. The results of these interactions depend on both the particles and the medium with which they interact. After an interaction, a part of the energy of the particle is often deposited into the medium in some way and this makes it possible to detect the particles.

1.1.1 Electrons and positrons

At high energies, the dominant interaction process for electrons and positrons is *bremsstrahlung*. When the charged particle moves through the electric field of another particle, typically the atomic nucleus, it will decellerate and the lost kinetic energy is converted into a photon. The energy loss by bremsstrahlung for electrons is given by

$$\frac{dE}{dx} = 4\alpha N_A \frac{Z^2}{A} r_e^2 \cdot E \ln \frac{183}{Z^{\frac{1}{3}}}$$

[1]. Here α is the fine-structure constant, N_A is the Avogadro constant, Z the atomic number, A the total number of protons and neutrons and r_e is the electron radius.

At lower energies, other processes like *ionization* start to play a role. At ionization, the electron will release a bound electron from the coulomb field of the nucleus. There are also other types of interactions like δ-ray production, Čerenkov radiation and excitation. Figure 1 shows that at energies above a few tens of MeV, bremsstrahlung is the only dominant process.

The *critical energy* E_c is sometimes defined as the energy at which the loss rate for bremsstrahlung is equal to the loss rate for ionization. Among alternate definitions is that of Rossi, who defines the critical energy as the energy at which the ionization loss per radiation length (see Section 1.1.3) is equal to the electron energy [2]. When using this definition, E_c is approximately given by

$$E_c = \frac{550\,\text{MeV}}{Z}$$

where Z is the atomic number [1]. Figure 1 shows that the critical energy, for lead, is a little less than 10 MeV. The particles of interest for the FoCal project range from a several GeV to a few hundreds of GeVs so bremsstrahlung is the most important process here.

1.1.2 Photons

As opposed to electrons and positrons, photons are massless and have no charge. Therefore they interact with matter in different ways. At high energies the dominant process is e^+e^- *pair production*. When a photon interacts with a nucleus, it can convert into an electron and positron. The cross section for the pair production of photons, which is related to the probability of pair production occuring, is given by

$$\sigma_{\text{pair}} \approx \frac{7}{9}\left(4\alpha r_e^2 Z^2 \ln \frac{183}{Z^{\frac{1}{3}}}\right)$$

[1]. At lower energies there are other interaction processes like compton scattering, the photo-electric effect and rayleigh scattering. These are not of interest however for the high energy photons that will be detected with the FoCal detector.

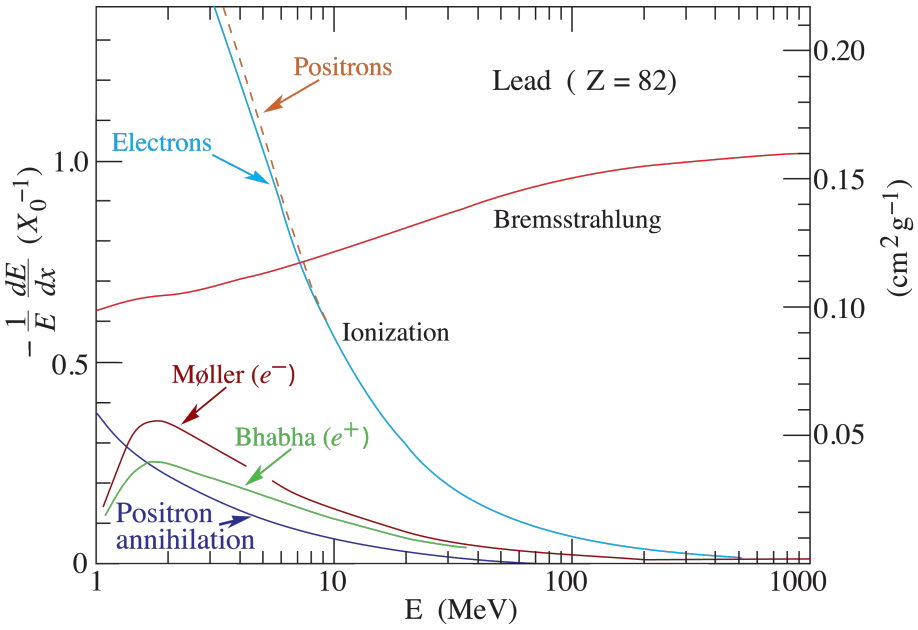

Figure 1: Fractional energy loss per radiation length in lead as a function of the electrons energy. Møller and Bhabha scattering in a medium lead to δ-ray production. Image taken from [2], page 19

1.1.3 Radiation length

As mentioned before, at high energies electrons mainly lose their energy by *bremsstrahlung*, and photons by e^+e^- *pair production*. This happens when the particles traverse through matter, and the characteristic length at which these interactions occur is called the radiation length, denoted by X_0.

The radiation length is defined as the distance over which an electron loses all but $1/e$ of its energy by bremsstrahlung. It is also $\frac{7}{9}$ of the mean free path for pair production by a high-energy photon. [2] This means a photon produces an e^+e^- pair after traveling a distance of $\frac{9}{7}X_0$ on average. When describing showers (see Section 1.1.4), X_0 is the appropriate length scale. Note that the radiation length depends on the materials through which the particles travel.

An approximation for X_0 is given by

$$X_0 = \frac{180A}{Z^2}\,\frac{\mathrm{g}}{\mathrm{cm}^2}$$

[1], where Z is the atomic number and A is the total number of protons and neutrons in the nucleus of the atom. This quantity is measured in $\mathrm{g/cm}^2$ so that it can be used to compare materials of different density ρ. Dividing by ρ will give the radiation length in centimetres.

The cross section of e^+e^- pair production by photons can be expressed in terms of X_0 by

$$\sigma_{\mathrm{pair}} \approx \frac{7}{9}\frac{A}{N_A X_0}$$

and the energy loss of electrons and positrons by bremsstrahlung can be rewritten to

$$\frac{dE}{dx} = \frac{E}{X_0}$$

[1]. This means $E = E_0 e^{-x/X_0}$ from which one can see that after one radiation length, the electron has only $1/e$ of its energy left.

The radiation length of a mixture of several materials can be calculated by

$$\frac{1}{X_0} = \sum \frac{w_j}{X_j}$$

where w_j is the fraction of weight and X_j the radiation length of the j-th material. The tungsten used in the FoCal detector has a radiation lenght of 3.5 mm. Combining this with the other materials of the detector, a radiation length of about 0.6 cm is obtained. [3]

1.1.4 Showers

When a high-energy electron or photon is incident on a thick absorber, pair production and bremsstrahlung will generate more electrons and photons with lower energy. This will cause a chain reaction and is also known as an *electromagnetic shower*. In such a shower, particles are constantly being created and destroyed. At the start of the shower, the number of particles rises due to the high number of creations until at some point, called the *shower maximum* the energy of the particles has dropped so far that more particles are being destroyed rather than created.

The longitudinal development is governed by the high-energy part of the shower and scales as the radiation length in the material. When the electron energies fall below the critical energy E_c, they will dissipate their energy by ionization rather than by the generation of more particles.

1.1.5 Molière radius

The transverse development of electromagnetic showers is described by the Molière radius, denoted as R_M. It is given by

$$R_M = X_0 \frac{21\,\text{MeV}}{E_c}$$

For a compound material, the Molière radius is given by

$$\frac{1}{R_M} = \frac{1}{21\text{Mev}} \sum \frac{w_j E_{cj}}{R_{M,j}}$$

where w_j is the weight fraction of the element with critical energy E_{cj}. [2]

The importance of the Molière radius lies in the following. On average, 90% of the deposited energy lies inside the cylinder with radius R_M, and about 99% is contained within $3.5\,R_M$. The value of this radius depends on the materials used and for the FoCal detector it is approximately one centimetre.

1.2 Detection specifics for FoCal

The FoCal detector is an electromagnetic calorimeter. It will be used to measure particles at high rapidities, meaning particles with a small angle to the beam pipe, so in a forward direction. Two options are being considered for the placement of the detector. It will either be at 3.6 m or at 8.0 m from the interaction point. At a higher distance, particles with higher rapidity can be measured which is preferred.

The detector is a *sampling calorimeter*, meaning it consists of an active medium that contributes to the signal and a passive medium that does not. The passive medium is meant to cause interactions and absorb energy. The material for this passive medium will be tungsten, because of its small radiation length: tungsten has a radiation length of $X_0 = 3.5\,\text{mm}$ and a Molière radius of $R_M = 9\,\text{mm}$. The material used for the active medium is silicon. [3]

The FoCal detector will consist of different segments, divided into different layers. A schematic overview of the currently proposed detector is given in Figure 2.

There are low granularity layers, also called coarse layers, where each cell has an area of approximately $1\,\text{cm}^2$. A measurement with these cells will return an analog value that is directly related

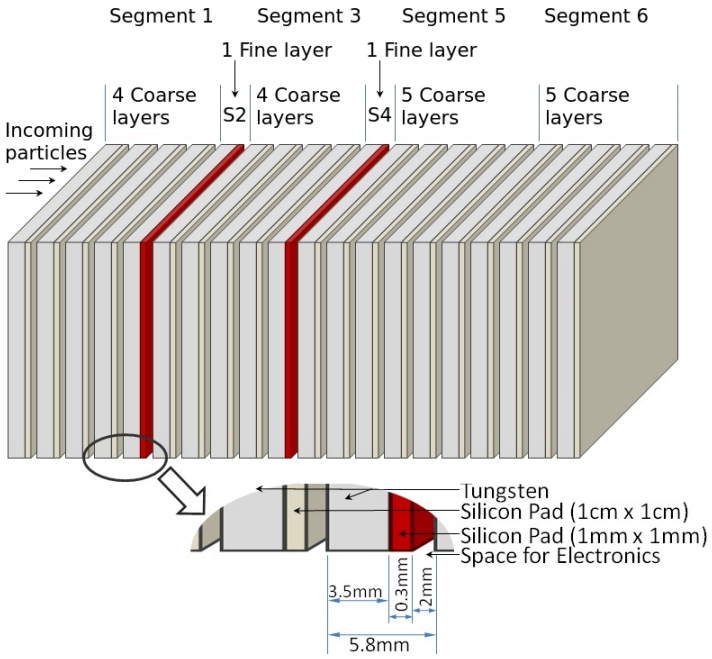

Figure 2: Schematic overview of the setup of the FoCal detector. Picture made by G. Zomer.

to the deposited energy in that cell. The responses of these layers are summed into segments on a hardware level (in the current proposal), so the software will only have access to the full segments.

The detector will also have high granularity layers, also known as fine layers, which are needed to separate nearby particles. These layers will consist of approximately 0.1 mm wide micropixels that can be either enabled or disabled, meaning they contain signal above a threshold or not. These micropixels are grouped in squares of 10 by 10 to obtain pixels (with dimensions $1 \times 1\,\mathrm{mm}^2$) that have an *amplitude* between 0 and 100 which is the amount of enabled micropixels. It is not known yet how small the actual micropixels will be, but they will probably be smaller than 0.1 mm. The software will only be able to access the result of the pixels, not the individual micropixels.

This combination of different type of layers should allow a high position resolution while keeping the costs down, as the high granularity layers are relatively expensive.

1.3 Simulation

For simulating particle showers, the GEANT software was used. This software can do Monte Carlo simulations of the passage of elementary particles through matter. Given a a list of materials with all their properties and their location, GEANT is able to simulate all the interactions that are known for the simulated particles.

The detector setup that is currently being used for these simulations consists of 6 segments in total (see Figure 2), placed at a distance of 360 cm from the interaction point. The second and fourth are high granularity (fine) segments and the others are low granularity (coarse) segments.

As mentioned before, the high granularity segments consist of 1 mm wide pixels (consisting of 100 micropixels) that have an amplitude between 0 and 100. For the simulation it is assumed that the summing of the micropixels happens in the detector hardware so the state of the individual micropixels can not be retrieved in the algorithm.

The low granularity segments consist of 1.0 cm wide pads that measure energy directly and return a numerical value. The values of both type of segments have to be scaled and calibrated in order

to obtain the actual deposited energy. In the real detector, the low granularity segments consist of multiple layers, but for the simulations these layers were summed and treated as a single segment.

The simulations used in this thesis do not incorporate any form of noise and there is no space between the cells. Furthermore all cells are *perfect* meaning that they always work and the measured value they return is always directly related to the deposited energy in that cell. This means it is a perfect detector in some sense, so any efficiency obtained with this simulation can not be guaranteed for the real detector. The data obtained by measurements with the real detector can have larger uncertainties and there is noise involved. There will be dead chips and other faults in the hardware which might be a fundamental issue that can not be solved directly by improving the hardware. Therefore the final algorithm for the real detector will need another step to process the data before the clustering algorithm is applied. All the research done for this thesis will be on the simulated *perfect detector*.

1.4 Rejecting π^0 decay products

One of the goals of FoCal is to measure the *direct photons* that are the result of parton interactions. A dominant fraction of the *other* photons are the photons resulting from a neutral pion. The π^0 particle decays into two photons ($\pi^0 \rightarrow \gamma + \gamma$ is 98.8% of π^0 decays, see [4]) and these photons are *not* the ones of interest so they must be recognized. At high energies these photons hit the detector under a small opening angle: they can be less than a centimetre apart. Once the photons are successfully detected, their energy and position information can be used to calculate the invariant mass of the system of two photons. This can then be used to check if two photons originated from a π^0 particle.

1.4.1 Invariant mass

The *invariant mass*, also called *rest mass* of an object is a characteristic of the total energy and momentum of the object. The invariant mass has the same value in all frames of reference which are related by Lorentz transformations. In the center of momentum frame of the object (if it exists) the measured energy of the object is minimal and the mass is simply given by $m = E/c^2$. When the object is measured from another frame of reference the measured energy is higher and the momentum of the object is subtracted so that the resulting mass is indeed invariant. From any frame of reference the invariant mass can be calculated from the particles energy E and its momentum $\vec{p}$ with the *energy-momentum-relation*:

$$m^2c^2 = \left(\frac{E}{c}\right)^2 - \|\vec{p}\|^2$$

Note that a rest frame does not exist for single photons and therefore their mass is zero. However a system of two photons *not* going in the same direction does have a center of momentum frame and therefore such a system does have an invariant mass. When using the four-momentum vector notation, the invariant mass can be found by calculating the magnitude of the vector using the Minkowski norm:

$$m^2c^2 = \left\| \left(\frac{E}{c}, \vec{p}\right) \right\|^2 = \left(\frac{E}{c}\right)^2 - \|\vec{p}\|^2$$

In particle physics it is common to use *natural units* which means $c = 1$. This simplifies the energy-momentum-relation to simply

$$m^2 = E^2 - p^2$$

With this system, the energy is often given in GeV, momentum is given in GeV/c and mass in GeV/c^2. This convention will be used in the following formula's.

When a π^0 particle decays into two photons, the system being considered is the system of both photons. Since energy and momentum are conserved during a decay, the invariant mass of the pair of photons is equal to the mass of the π^0 particle which is a known constant. This makes the invariant

mass a useful tool to check if two clusters are two photons resulting from a π^0 decay instead of being direct photons.

Assuming that the energy and position of two measured particles are calculated correctly, the invariant mass of the pair can be calculated. Let $\vec{r} = (x, y, z)$ be the location where the particle hit the detector and E be the measured energy. The photon travels from the ogirin to the point $\vec{r}$ so therefore its momentum will have the same direction. Since the particle is a photon the magnitude of the momentum vector is given by E. This can be seen directly from the energy-momentum-relation mentioned above by setting $m = 0$ for a massless particle. Combining this, the momentum vector of the photon is given by:

$$\vec{p} = E\frac{\vec{r}}{\|\vec{r}\|} = E\hat{r}$$

The four-momentum vector is then given by

$$(E, \vec{p}) = E\,(1, \hat{r}) = E\left(1, \frac{x}{\|\vec{r}\|}, \frac{y}{\|\vec{r}\|}, \frac{z}{\|\vec{r}\|}\right)$$

The length of the vector (using the Minkowski metric) is zero which is correct because a single photon does not have invariant mass. However when two particles are detected, at different locations $\vec{r}_1$ and $\vec{r}_2$ with energies E_1 and E_2 then the four-vectors can be added resulting in a vector with a non-zero norm:

$$(E_1 + E_2\,,\, E_1\hat{r}_1 + E_2\hat{r}_2)$$

This summed vector corresponds to the system of the two particles and the invariant mass of that system can then be found by calculating the Minkowski norm of the summed vector. This can be rewritten to express the mass in the opening angle θ between the two photons:

$$\begin{aligned} m^2 &= \left((E_1 + E_2)^2 - \|E_1\hat{r}_1 + E_2\hat{r}_2\|^2\right) \\ &= \left(E_1^2 + 2E_1E_2 + E_2^2 - \left(E_1^2 + 2E_1E_2(\hat{r}_1 \cdot \hat{r}_2) + E_2^2\right)\right) \\ &= 2E_1E_2(1 - \hat{r}_1 \cdot \hat{r}_2) = 2E_1E_2\,(1 - \cos\theta) \\ m &= \sqrt{2E_1E_2(1 - \cos\theta)} \end{aligned}$$

1.4.2 Invariant mass at multiple clusters

At a real measurement many particles will be detected at once resulting in many clusters. It can not be known which of the clusters form a pair that originated from a single particle so it is not clear of which cluster pairs the invariant mass should be calculated. Therefore, the invariant mass of *every pair of clusters* will be calculated. A frequency distribution of these invariant masses should then show a peak at the π^0 mass, and at the rest masses of other particles. If the invariant mass of a pair of clusters is close enough to that of the π^0 mass then the pair can be marked as a π^0 pair. If the detector has enough accuracy this should recognize most of the π^0 particles so they can be subtracted as background signal.

2 Clustering

2.1 Problem description

Clustering, or cluster analysis, is a general term not only applied in particle physics. Clustering is the task of grouping data points in such a way that points in the same group have some kind of similarities, whereas points in different groups do not have these. There is no specific algorithm for clustering as it depends on the type of data and on the kind of similarities that data points in the same cluster need to have.

One can make a distinction between hard and soft clustering. In hard clustering, also known as *crisp* clustering, each data point belongs to exactly one cluster. In soft clustering, or *fuzzy* clustering, a data point can belong to different clusters each with a certain probability.

The type of clustering algorithm also depends on prior knowledge of the data points or knowledge of cluster properties. An example of this can be an expected number of data points in a cluster or a maximum distance of a data point to a cluster (where distance can be any function that represents a measure of similarity).

To further complicate things, the data points can have uncertainties caused by the limited precision of a measuring device. There can also be noise involved which is the case for any kind of detector. This means that it might not always be possible to perfectly clusterize the data but a good clustering algorithm will take all of this into account and still produce reasonable results.

2.2 General solutions

A very basic and general approach for clustering an n-dimensional grid of pixels (grid points can be enabled or disabled) would be finding connected components. One could start at any enabled point and keep searching in its neighbourhood for enabled points thare are connected to it and merge those, repeating this until no more changes occur. One of the disadvantages of this method is that when two actual clusters overlap then this method will produce a single cluster.

When one has more information about the data, for example an expected cluster size and an expected shape (a circle for example) then the following procedure can be applied:

- Group pixels together in bins. The bin size incorporates knowledge of the expected cluster size.
- Search for local maxima (bins of which all neighbouring bins have lower counts) and define those as clusters.
- For each pixel, or for each bin, decide which local maxima it is closest to and assign it to that cluster

This method will work in simple scenarios and can already separate overlapping clusters to some extent. To further improve results one needs to incorporate more knowledge about the data points and cluster properties.

2.3 The FoCal scenario

Roughly speaking, the FoCal detector could be seen as a three dimensional grid. The cells of this grid are the objects that need to be clustered according to distance. Every cell has an amplitude that is related to the deposited energy in that cell and this amplitude is also incorporated into the clustering. The full detector setup, as used in the simulations, was explained in Section 1.3.

The clustering method presented in the previous section is by far too simple for the FoCal detector. The general idea is still valid and is used in the clustering algorithm but many additional steps are applied. Most of these steps are based on specific features related to the physics of particle showers. These features can be used to distinguish clusters that overlap which would not be possible without prior knowledge about the data.

2.4 Description of the full algorithm

The algorithm that is currently used for the FoCal detector is split into two parts: first constructing clusters at each segment of the detector and then combining those results to reconstruct a full cluster at the level of the detector.

2.4.1 Cluster algorithm definitions

The **amplitude** is the detected signal of a single cell and it represents the deposited energy in that cell. All cells have a *flag* that specifies if it can be considered as a **seed** which means it is a possible cluster center. Each cell has a **weight** that is used to calculate the fraction of it belonging to a certain cluster. The amplitude of a cell is distributed amongst different clusters according to this weight.

2.4.2 Parameters

There are several parameters involved in both parts of the algorithm:

- ***MinRing*** specifies the minium distance between two clusters. No two seeds can be created within this distance from each other. In the current implementation the MinRing parameter has a value of 1 ring for the coarse layers, meaning there must be at least one cell (1 cm) between two clusters. The fine layers have this parameter set to 2 rings which means a distance of 2 mm between to clusters.
- ***MaxRing*** is the radius in which a seed can collect energy. It is assumed that the full particle shower will be contained within this distance. In the current implementation this distance is set to 5 cm for the coarse layers and 4 cm for the fine layers.
- ***SeedThreshold*** is the minimum amplitude that a cell needs to have in order for it to be considered as a possible seed. In the current implementation this threshold is not used (set to zero) because low energy clusters are already rejected at a later step of the algorithm by the ***ClusterEnergyThreshold***.
- ***NCellsThreshold*** is the minimum (weighted) number of cells that a cluster needs in order to be kept.
- ***ClusterEnergyThreshold*** is the minimum energy that a cluster needs. In case of the coarse segments this value is given in keV (for this simulation) but for the fine segments this value is being compared to the sum of cell amplitudes. These amplitudes are counts of micropixels that have signal above a threshold (on a hardware level) and this count is directly related to energy.
- ***Weight function***. This function describes the transverse profile of the cluster. This function and its goals are the main topic of chapter 3. The function is used to distribute the energy of a cell to different seeds. It is scaled so that the amplitude at the center of the function matches the amplitude at the center of a cluster. The weight function will give the expected amplitude at nearby cells. This value is used as weight for the neighbour to specify how much of that neighbour belongs to this cluster.
- ***RejectionRatio***. The amplitude of a neighbour cell is compared to the expected value (based on the weight function). If it is more than ***RejectionRatio*** times higher then the neighbour is energetic enough to possibly become a separate cluster. This parameter is explained in chapter 3.

See table 1 for the values of these parameters in the current implementation.

Segment	MinRing	MaxRing	NCellsThreshold	ClusterEnergyThreshold	RejectionRatio
0 - coarse	1 rings	5.0 cm	2.0	5000 keV	10
1 - fine	2 rings	4.0 cm	3.0	7	3.5
2 - coarse	1 rings	5.0 cm	3.0	10000 keV	10
3 - fine	2 rings	4.0 cm	10.0	10	3.5
4 - coarse	1 rings	5.0 cm	6.0	15000 keV	10
5 - coarse	1 rings	5.0 cm	5.0	8500 keV	10

Table 1: Parameter values for the clustering algorithm (May 2013). Note that for segment 1 and 3 the energy values are counts of micropixels that are hit, which is directly related to the energy.

2.4.3 Segment level algorithm

This section briefly describes the algorithm that is executed separately for each segment of the detector. The full version of this algorithm can be found in the appendix.

- The **SeedEnergy** is computed for each cell. It is the sum of the amplitudes of cells within (and including) the ***MinRing*** radius.
- The cells are sorted according to their amplitude (energy) and the algorithm loops over the cells from high to low amplitude in order to create **seeds**. Only cells with an amplitude above ***SeedThreshold*** are considered here.
  - The current cell (that is being looped over) is marked as a **seed**.
  - Every neighbouring cell that lies within the ***MaxRing*** radius is then inspected to see whether or not it can be part of a separate shower.
    - The neighbour cell belongs to this seed if its SeedEnergy matches the *expected* value, or when it is within ***MinRing*** radius. In this case the cell is marked so that it can no longer form a separate seed. The algorithm computes the expected value of the SeedEnergy of the neighbour based on the current cell's SeedEnergy, and a ***weight function***. If the neighbour is energetic enough, meaning its SeedEnergy is more than a factor ***RejectionRatio*** times the expected value, then this neighbour is left unchanged (not marked) so that the algorithm will find it later in the cell loop.
      The research done for this thesis focusses on optimizing the expected value and rejection ratio mentioned here. It will be explained in detail in Chapter 3.
  - All neighbours are given a **weight** based on the current cell's SeedEnergy. This weight indicates which fraction of the neighbour amplitude belongs to this seed.
- The list of seeds that has now been created is considered a second time to implement further rejection criteria.
  - The weighted number of cells is calculated: if a cell has weight contributions from multiple seeds then only the fraction belonging to the current seed is used. If the weighted number of cells is below ***NCellsThreshold*** the seed is rejected.
  - The total amplitude is calculated, again only taking the fraction belonging to the current seed. The seed is rejected if the total energy is below ***ClusterEnergyThreshold***.
- The remaining seeds will make up the final cluster list
  - The total energy is calculated the same way as in the previous step. This has to be repeated because some seeds may have been removed changing the weights.

– The mean position of the cluster is calculated. The amplitude of a cell is used as weight and the mean is taken over cells within 3 rings (inclusive) distance.

$$\vec{r}_{mean} = \frac{\sum E_i \vec{r}_i}{\sum E_i} \qquad \text{sum taken over digits within 3 rings}$$

Other ways of weighing, like using the logarithm of the amplitude, can give better results in some cases but that is not part of this research.

– The semi-major and semi-minor width of the cluster are calculated.

2.4.4 Detector level algorithm

At this part of the algorithm the resulting clusters of the previous section, at the segment level, are refered to as sub-clusters. These sub-clusters all have a flag specifying whether they have been **merged** and they have an energy **weight**, which is a variable used for this part of the algorithm and is seperate from the energy of the cluster measured in the first part. The merged flag and energy weight are reset when this part of the algorithm starts. This section will give a summary of the algorithm and the full description can be found in the appendix.

- First the segments are combined with other segments of the *same granularity* into semi-final clusters. This results is two lists of semi-final clusters: one for the low granularity segments and one for the high granularity segments.

  – The algorithm looks for clusters in the other segments of same granularity and merges them if they are less than ***MinRing*** distance apart. Energies are added and the position is calculated as an energy weighted mean.

  – The energy of the semi-final cluster is calculated according to the following formula:

$$E = p_0 \left(p_1 + \sum E_{sub} \right)^{p_2} + p_3$$

  where p_i are parameters that can be calibrated and E_{sub} are the energies of the sub-clusters. The values used in the current implementation are given in Table 3 in the appendix. If the resulting energy is not positive then the cluster is not saved.

- For each high granularity semi-final cluster, the closest low granularity semi-final cluster is taken (within ***MaxRing*** distance). This creates a final cluster where the position information is taken from the high granularity semi-final cluster, and the energy information is taken from the low granularity cluster.
  If a low granularity cluster was used by multiple high granularity clusters, the energies of the high granularity clusters are used as weights to divide the energy of the low granularity cluster amongst them.

- If there are low granularity semi-final clusters that are not used yet then they also create final clusters but with no high granularity information. The energy and position information are taken directly from the low granularity semi-final cluster.

This completes the second part of the algorithm.

2.5 Cluster properties

The final clusters that are the output of the clustering algorithm have the following information:

- position
- total energy and energy for each coarse segment
- semi-minor width and semi-major width per segment

The particle showers are symmetric and should therefore produce round clusters. If the shape of the cluster is too elliptic (based on the semi-minor and semi-major width) then this is a sign that the cluster could be caused by two particles very close together instead of a single particle. Note that for the low granularity segments with 1 cm pads, calculating the widths is a hard task because the resulting shape depends greatly on where the particle hits the pad. Using this shower shape in order to distinguish particles is currently not implemented in the algorithm but it might be used in the future.

A property of every *pair* of clusters is their invariant mass, which can be used to decide whether the two particles originated from a π^0 decay as explained in Section 1.4.1.

3 Transverse shower profile

The algorithm in Section A.2 mentions a *weight function*. It is this function that will be the focus of this chapter and from now on it will also be referred to as the *profile function*. First its physical interpretation is explained, followed by the goals of the function for the clustering algorithm.

3.1 Notation

At this point it is important to clear up the notation that is used for (density) distributions. The following holds for any density but in this case the deposited energy E will be used. When considering an energy distribution along the x-axis, the energy density is denoted as $\frac{dE}{dx}$. With this notation the variable E is a cumulative quantity so that $E(x)$ can be seen as the total energy from $-\infty$ to x and with that in mind $\frac{dE}{dx}(x)$ will be the density at point x.

When generalizing this to a two-dimensional distribution the density is denoted as $\frac{d^2E}{dxdy}$. However when one wants to think of $E(x, y)$ as a cumulative variable, it is not well-defined of which region this is. For particle showers the distribution should be radial, meaning it only depends on the distance r to the center. When one wants to describe the density as a function of r there is an important difference caused by the Jacobian of a polar coordinate transformation. When transforming the cartesian coordinates (x, y) to polar coordinates (r, φ), an infinitesimal area $\mathrm{dx}\,\mathrm{dy}$ transforms to $r\,\mathrm{dr}\,\mathrm{d}\varphi$. This means $\frac{d^2E}{dxdy}(x, y)$ is equal to $\frac{d^2E}{rdrd\varphi}(r, \varphi)$ and when the distribution does not depend on φ this can be simplified to

$$\frac{1}{2\pi r}\frac{dE}{dr}(r)$$

This quantity will then equal the density at a point with a distance r to the origin. To obtain the cumulative energy within the region with radius R, the integral over this region must be taken and one finds

$$\int_0^R \int_0^{2\pi} \left(\frac{1}{2\pi r}\frac{dE}{dr}(r)\right) r d\varphi dr = \int_0^R \frac{r}{r}\frac{dE}{dr}(r)dr = E(R) - E(0)$$

So with this notation $E(r)$ can be viewed as the cumulative energy within distance r to the origin (if $E(0)$ is set to zero).

When showing a radial energy distribution, $\frac{1}{2\pi r}\frac{dE}{dr}$ is shown instead of $\frac{dE}{dr}$ because this gives a better visual idea of the distribution. However when one wants to generate random events according to the distribution by randomly drawing values for r and φ, then values for φ can be drawn according to a uniform distribution and values for r have to be taken from the distribution $\frac{dE}{dr}$ instead of $\frac{1}{2\pi r}\frac{dE}{dr}$.

3.2 Physical interpretation

The physical interpretation of the profile function is that it describes the transverse profile of the shower. An example of such a distribution can be seen in Figure 3. The analytical form of the actual distribution function is not known, since the showering of the particle consists of many different processes with different probabilities. Different functions have been fitted to measured and simulated data.

In a lecture on particle detectors [1] it is suggested that the distribution is of the following form:

$$\frac{1}{2\pi r}\frac{dE}{dr} = \alpha e^{-r/R_M} + \beta e^{-r/\lambda_{min}} \tag{1}$$

where α, β are free parameters, R_M is the Molière radius and λ_{min} is the range of low energetic photons. The document suggests that the outer part (larger values of r) of this function is caused by low energetic photons which implies $\lambda_{min} > R_M$. However when peforming fits on simulated data (single gamma events with $0.5\,\mathrm{GeV/c} \leq p_T \leq 20\,\mathrm{GeV/c}$) one finds that the same function with

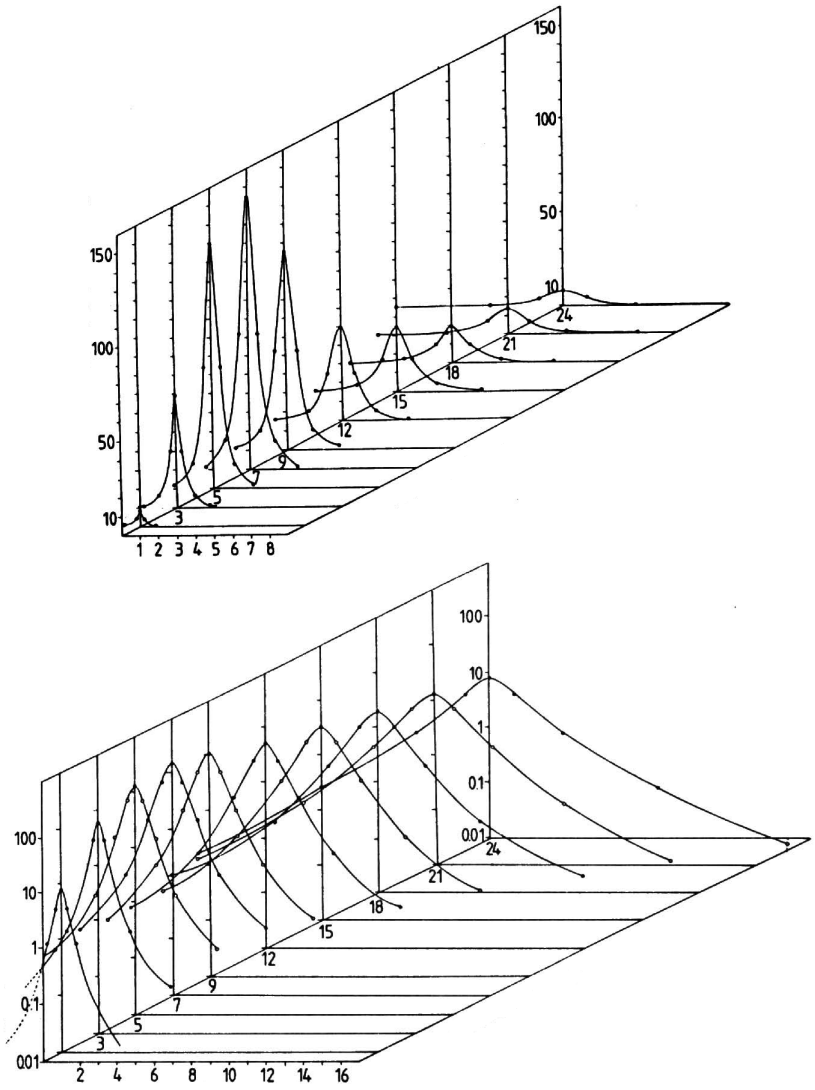

Figure 3: Longitudinal and lateral electromagnetic shower profile for a 6 GeV electron in lead. The top picture shows the distribution on a lineair scale and the bottom picture shows the same profile on a logarithmic scale. The numbers on the horizontal axis and the depth axis both represent multiples of the radiation length X_0. The vertical axis shows the energy deposit in arbitrary units. Source: [1]

$\lambda_{min} < R_M$ matches the data significantly better. In this thesis the function (1) is referred to as the *double exponential.*

The distribution function that is implemented as the weight function in the algorithm was given by

$$\frac{1}{2\pi r}\frac{dE}{dr} = \frac{I}{\left(\frac{r}{\sigma_0}\right)^2 + e^{r/\sigma_1}} \tag{2}$$

where σ_0, σ_1 are two parameters that specify the width of the shower and I is the energy or intensity. This function is a modified Cauchy distribution (also called Lorentz distribution) that is often used in (particle) physics:

$$\frac{I}{1 + \left(\frac{r}{\sigma_0}\right)^2}$$

The exponential part was added to make the function tend to zero faster at large r. This thesis will refer to the function (2) as the *damped cauchy function.*

3.3 The profile function in the algorithm

In the algorithm the profile function is used for different purposes. Recall the definition of **seed energy** as mentioned in the algorithm: the seed energy of a cell is the sum of the amplitudes of cells within **MinRing** radius. When a new seed is found, the profile function is scaled so that it matches the seed energy of the seed cell at $r = 0$. The value of the profile function (at different values of r) will then give an estimated value for the seed energy of neighbouring cells. It must be noted that the transverse shower profile as described previously refers to the energy deposited at different positions which is directly related to the amplitude. The seed energy however is a little different from this. The distribution of seed energies could be seen as a smoother version of the energy distribution.

The first reason for comparing seed energies instead of amplitudes is to counter fluctuations. The amplitude of a single cell can be far above the average, especially at higher distances to the cluster center (3 cm). At the smoother seed energy distribution these fluctuations are compensated for to some extent.

The large pads in the low granularity segments give another reason for using the seed energy. The cells in these segments are 1 cm wide and this is very close to the Molière radius of the showers that need to be detected. This means that when a particle hits the detector in the center of such a cell, almost 90% of the shower's energy will be deposited within that cell. The resulting cell amplitude distribution will have a single large peak. However when a particle hits the detector on the edge of two such cells then its energy will be divided among these two cells and the amplitude distribution will have two peaked cells instead of one. Using the seed energy will compensate this effect because the sum of the amplitudes of the two peaks should be almost equal to the amplitude of the single peak. The seed energy distribution of the two different cases should therefore more similar whereas the amplitude distribution differs depending on where the particle hits the cell. The same effect is present when the particle hits the cell under an angle, and also here the seed energy distribution will compensate the effect to some extent.

3.3.1 Energy assignment

The first usage of the profile function assigning energy to different clusters. When a new seed is added, the *expected* seed energy for a neighbour is calculated and this value is added to the *weight* variable of the neighbour. When all seeds have been processed, all cells will have some non-zero *weight* which can be the sum of weights by different seeds. The fraction of this weight that belongs to a seed is the fraction of the amplitude that will be assigned to that seed.

This weight function should resemble the previously discussed transverse shower profile and the reasoning for that is as follows. If n particles hit the detector, and the energy deposition of particle i at position x, y is given by $f_i(x, y)$ then the total energy deposition at position x, y is simply the sum$\sum_{i=1}^{n} f_i(x, y)$. When the particles are of the same type the energy profile should only depend on the distance to the particle hit location (because the shower is symmetric) and on the energy of the particle. When multiple particles hit the detector close to each other, the energy deposition of the single particles is not known (only their sum) so this must be estimated which is what the weight function does. As explained, the weight function depends on the seed energy at the center and on the distance from a neighbour to that center, so it can be written as $g(E_i, |\vec{r} - \vec{r}_i|)$. The algorithm currently uses the modified cauchy distribution where the energy is a scalar: $g(E_i, |\vec{r} - \vec{r}_i|) = E_i \tilde{g}(|\vec{r} - \vec{r}_i|)$. The fraction of the total deposited energy at position $\vec{r}$ that belongs to particle i is then given by

$$\frac{g(E_i, |\vec{r} - \vec{r}_i|)}{\sum_{j=1}^{n} g(E_j, |\vec{r} - \vec{r}_j|)}$$

Note that when the weight function is multiplied by a scalar, the resulting fractions remain the same so only the shape of the function is important.

3.3.2 Seed rejection

The second purpose of the profile function is rejecting seeds. The algorithm starts by considering the cell with the highest amplitude and uses that as a first seed. Cells in the neighbourhood of that seed are then rejected as possible seeds based on the profile function. The expected seed energy of the neighbour is calculated and if the actual seed energy is more than a factor ***RejectionRatio*** (see table 1) higher, it can be considered as a separate seed (i.e. not rejected here). If the seed energy of the neighbour is not high enough, the cell is rejected, meaning it can no longer be a seed. The rejection ratio is based on the statistical fluctuations of the seed energies. It must be noted that the rejection ratio does not depend on r (the distance to the center of the cluster) so this assumes that the fluctuations in seed energy, as a ratio of the average amplitude, are the same at each distance.

The main topic of this research is optimizing the profile function and rejection ratio at the rejection step of the algorithm. When trying different functions and parameter values, the weight function used for assigning energy was left unchanged to make sure there were no side effects caused by wrong energy values. Since the rejection function is now different from the energy weight function, the rejection ratio parameter was ignored (set to 1.0) and this value was incorporated into the rejection function itself by scaling it.

4 Results

This chapter will discuss the different methods that have been tried to improve the rejection function.

4.1 Measuring efficiency

At every attempt for improving the rejection function, the algorithm was executed on 10,000 simulated single gamma events and 10,000 single π^0 events. Both gamma and π^0 events were simulated with transverse momenta ranging from $0.5\,\mathrm{GeV}/c$ to $20\,\mathrm{GeV}/c$ (uniform). The single gamma events should produce exactly one cluster, whereas the π^0 decays into two gammas and should produce two clusters. This is unless one of the photons goes outside the acceptance range of the detector but that only happened for 0.3% of the events so it can be ignored.

The peformance of the algorithm, after running it with different parameters, was evaluated by considering two variables. The first quantity that is considered is the number of clusters found by the algorithm. The second variable is referred to as *efficiency* in this thesis. For each event it is checked whether there is a cluster present at the simulated particle incident location. If all simulated particles (one for gamma events, two for π^0 events) have a cluster within 0.135 cm distance to its real location then a flag is set to true. The *efficiency* is defined as the percentage of events that have this flag set to true.

It is important to note that when the cluster algorithm finds too many clusters, there is a large probability that some of those clusters will be at the simulated particle positions and so this efficiency will then be very high. If every cell would be considered as a cluster for example, the efficiency would always be 100%. Therefore the total number of clusters that was found must also be taken into account. In Section 4.3 it will be explained that it is better to find more clusters than finding fewer clusters.

4.1.1 Single gamma efficiency

In Figure 4 the results of the algorithm with the original (unmodified parameters) Cauchy rejection function can be seen for single gamma events. The resulting efficiency is very high and the goal is to maintain at least this level of results for single gamma events while improving the results of π^0 events. When trying different rejection functions it turned out that the results of the single gamma events did not change significantly and the efficiency was always high enough. Therefore, when comparing the results of different rejection functions in later parts of this thesis, only the results of π^0 events are shown.

4.2 Optimizing the profile function

To obtain reasonable parameters for the profile function one needs to have an idea of how a characteristic single particle cluster looks. Therefore 10,000 single gamma particle events were simulated and this results in 10,000 samples of a 6 grids (one for each segment) with amplitudes. The profile function can then be chosen to match these samples. To do this, fits were done on these samples as will be explained below. In every case both the *damped cauchy function* and the *double exponential* were fitted to the data points. Note that the simulated events have a wide energy range and one could expect the shape of the profile to depend on the energy, whereas the current functions only depend on the energy via a scale factor. This potential issue will be addressed in Section 4.2.5 after seed energy distributions and the concept of rings have been explained.

4.2.1 Average amplitudes

To start with, the different profile functions were fitted to the ampitude distributions of single gamma events. Figure 5 shows an example of such fits. The data is a two-dimensional grid of cells but since these are difficult to visualise the x-projections of the grids are shown. By doing this projection the type

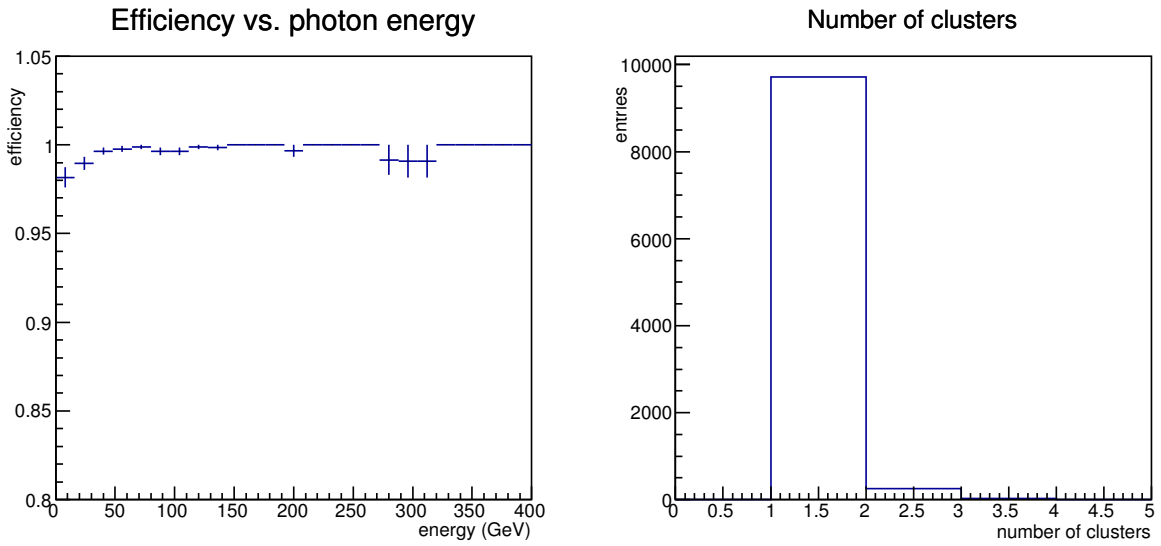

Figure 4: Results of the clustering algorithm for 10,000 single photon events using the original damped Cauchy rejection function. The definition of efficiency is explained in Section 4.1.

of distribution changes because the distribution is not of the form $f(x, y) = g(x)g(y)$ so the resulting fit parameters cannot directly be used. The fit however is also performed on the two-dimensional data set and these projections are done only to visualise how well the functions fit the data. It can be seen that the fluctuations, especially in the fine segments, are large for this particular shower. In segment 2, for example, a peak is present on the left and one might expect this to be a separate particle with lower energy but it is not. These fluctuations are also the cause of the poor fit results. Note that the graphs have a logarithmic scale and at some points the fit deviates from the simulated amplitude by more than 10 times its value so the resulting fit parameters are not very useful.

To overcome this effect different events were added together, after translating them so the centers of the clusters overlap, to obtain the average profile of multiple events. For every event the amplitudes were first scaled so that the amplitude at the cluster center was equal to one. After this scaling, the average was taken over 10,000 events and the results can be seen in Figure 6. The error bars show the root mean square of the values (instead of the error of the mean), in order to visualise the shower fluctuations. Even though segment 4 en 5 show higher values, the amplitudes in these segments are not higher because all distributions were scaled. It only shows that the distribution in these segments is less peaked. Since most of the segments are very similar, the figure only shows segments 1, 2 and 5.

The figure also shows the damped cauchy weight function that is used to assign energy to seeds (in green) and a fitted version of that function (in red). The original unfitted cauchy weight function does not seem to match the profile well at the first four segments and the fitted version does not seem to improve this. However, as explained previously, multiplying the weight function by a scalar does not influence the results of weighting cells. Since the plot has a logarithmic y-axis, a multiplication is a translation in this plot. The weight function, after translation, does match the shape of the data. As explained in Section 3.3.2, the energy assignment weight function was separated from the rejection function so it is not very relevant for this research.

One can see that the average shower profile gives better fit results than fits on single events. However the error bars show that the amplitudes of single events can differ from the average by large amounts. This effect should be incorporated in the rejection function as will be explained later.

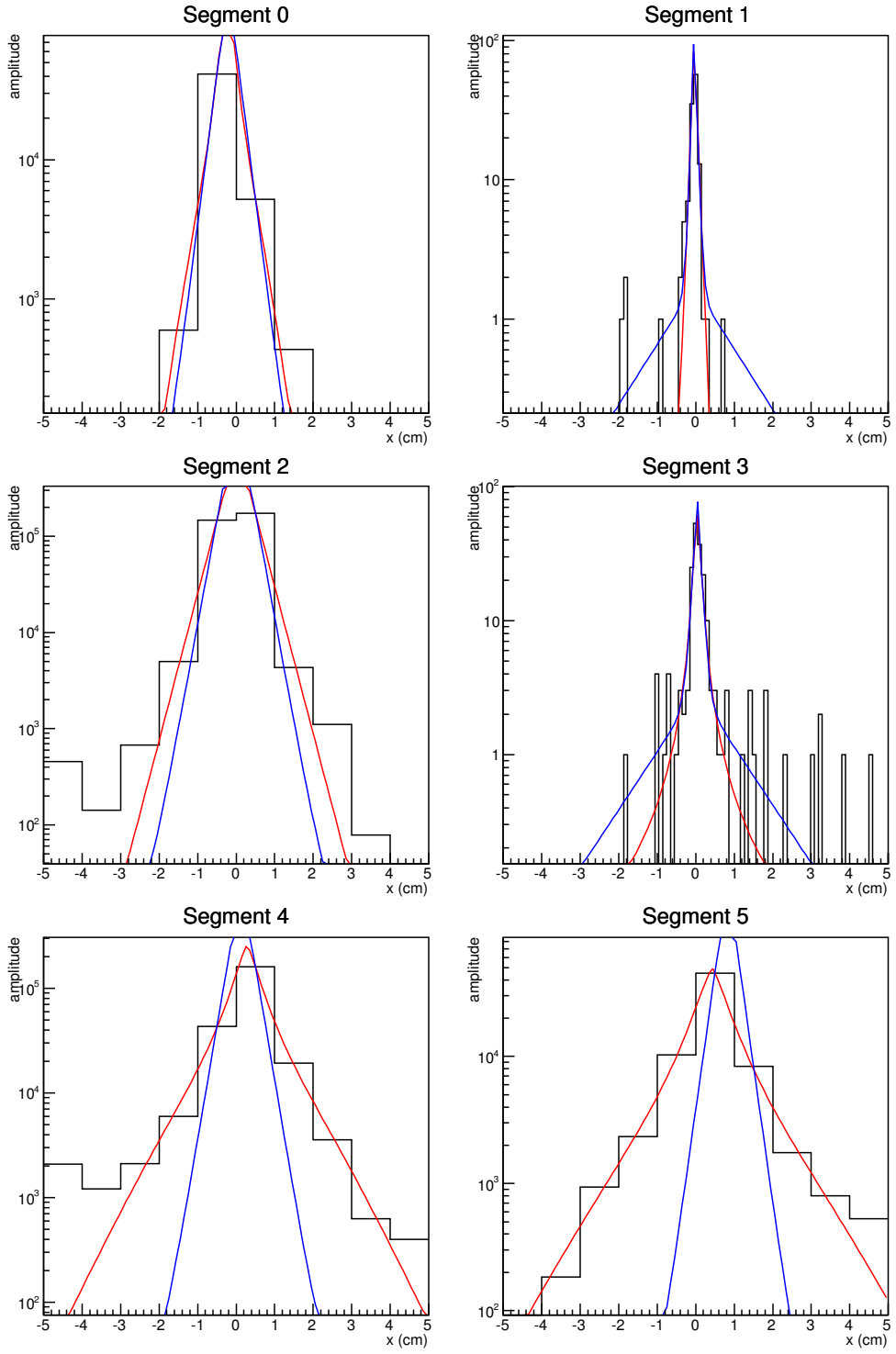

Figure 5: Fit of different profile functions on a **single** gamma event. The blue line is a fit of the *double exponential* and the red line is a fit of the *damped cauchy function* (defined at the start of chapter 3). The plots show the x-projections of the actual two-dimensional data.

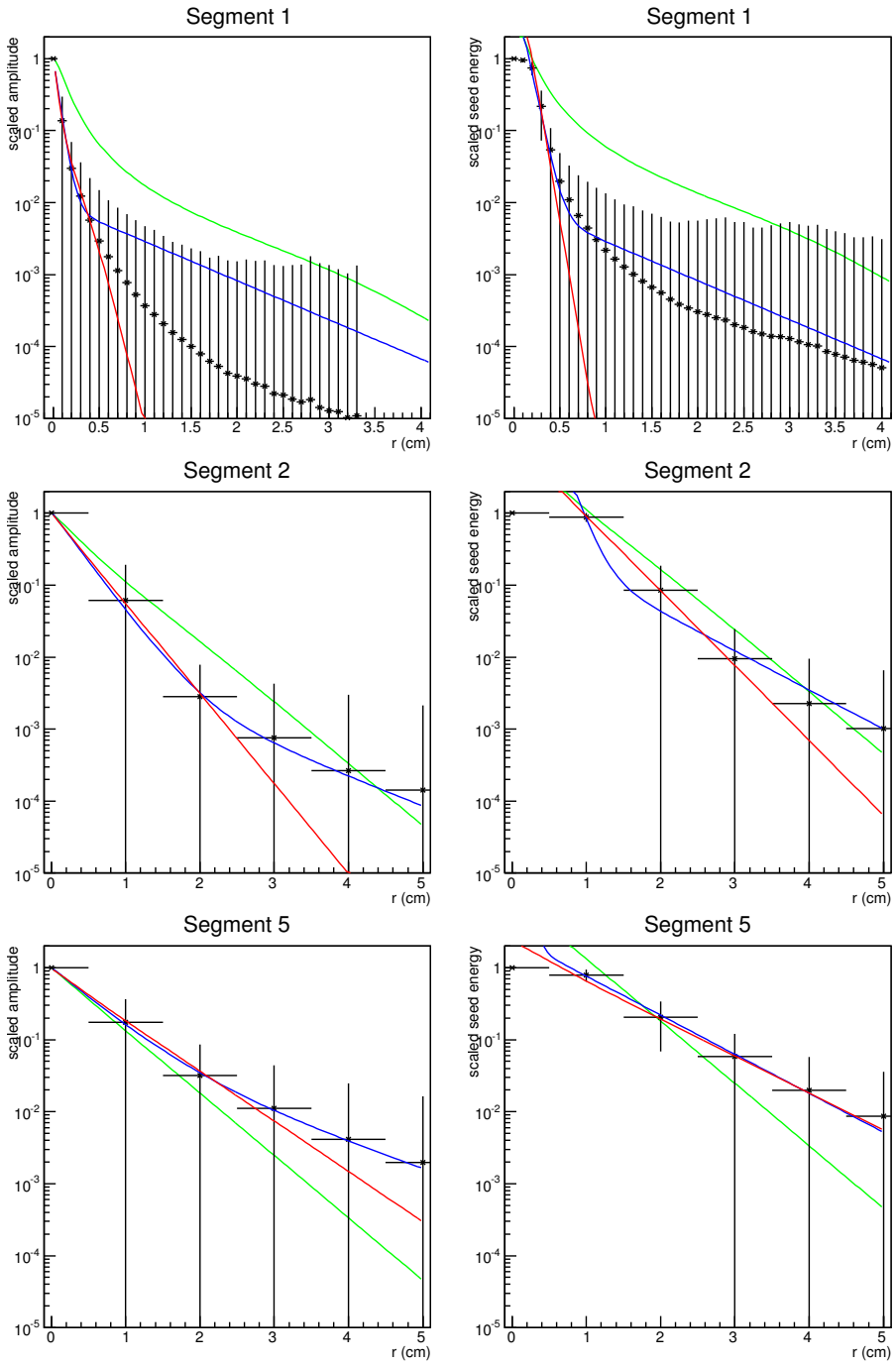

Figure 6: Fit of different functions on the average of 10,000 single gamma events shown as a radial distribution: $\frac{1}{2\pi r}\frac{dE}{dr}$. The error bars show the root mean square of the values. On the **left** the average *amplitudes* are shown and on the **right** the average *seed energies* are shown. The blue line is the *double exponential* fit, the red line is the *damped cauchy* fit and the green line is the damped cauchy weight function (not fitted) that is being used for assigning energies to seeds. The green function on the right includes the rejection ratio, so it is the function used for seed rejection. The vertical axis represents deposited energy but at every event, at every segment, the amplitudes or seed energies were scaled as explained in Section 4.2.1 and Section 4.2.2

4.2.2 Average seed energies

As mentioned before, the algorithm compares the *seed energies* of different cells instead of amplitudes. Therefore the radial distribution seed energies was calculated, similar to the amplitude distribution in the previous section. For different events the seed energy distribution was then scaled so that the cell at the center had a seed energy equal to one. After scaling, the average was taken over all events and the fits were then performed on this average. Since the algorithm is comparing seed energies, the results of this fit should be more appropriate for a rejection function. The fit result can be seen in Figure 6. It must be noted that the amplitude distribution is sharply peaked and in a coarse segment a large portion of the energy is often deposited in a single cell. However when calculating the seed energies, the value of this single center cell is also added to all eight neighbours. This results in a square of nine cells with very high seed energies and this is seen in the figure by the high value of the first bin next to the origin. Since the damped cauchy and double exponential functions are very peaked, this effect causes bad fit results near the origin. In Section 4.2.6 it is explained how the fits are repeated on a different range. As a result of the flattening effect of the seed energies, this method should not be used to find the *weight function* that assigns energy since this distribution does not describe a physical quantity. The clustering algorithm does use the seed energies at the rejection process however and therefore it is the right quantity to consider here, but it should be noted that the rejection function is now very different from the amplitude weight function.

4.2.3 Fluctuations

The fits done so far are all based on average values. If the resulting function would be used directly as a rejection function (without any additional rejection ratio) then every cell that fluctuates above the average would be considered as a separate seed. This behaviour is not desired and it is therefore necessary to find a function that does not describe the average seed energies but a good upper bound of the seed energy fluctuations. The rejection function should be chosen so that, when the seed energy of a neighbour cell is higher than the rejection value, the chance that it belongs to the same cluster (as the seed) is, for example, less than 1 percent. To characterize the fluctuations (whether they are Gaussian and so on), the distribution of amplitudes at a fixed distance from the cluster center was computed.

The detector consists of square cells in a grid and therefore does not have all the same symmetries that the showers have. When searching the neighbourhood of a cell, the algorithm loops around the grid in so-called *rings*. Figure 7 shows how the rings are defined in the grid. Ring zero is the center cell and ring one is the 8 cells in a square around it.

To start with, for each event, for each segment, for each ring, the maximum seed energy on of the cells on that ring was calculated. The maximum is taken over the whole ring, per event, per segment. This was done for 10,000 events and for each ring, for each segment, the average over all events was calculated. Fits for the rejection function were then done on this average maximum profile. The results of segment 1 and 2 can be seen in Figure 8. The other segments were similar and therefore not shown.

The data of segment 1 shows that the damped cauchy fit is very poor, due to the high values in the first two rings. The double exponential function seems to match the data better, in both segments. The plots also show that the original damped cauchy function, compared to the average maximum, might have been too high at low range and too low at high range, suggesting that the exponential term in this function might not be necessary. To find out whether this average maximum is a good indication for the rejection function, more insight into the fluctuations is needed.

In order to learn more about the fluctuations, the amplitude and seed energy distribution per ring was considered. For each segment, for each ring, the frequency of each amplitude and seed energy occurring in a ring was saved. The amplitude distribution for some rings can be seen in Figure 9. The plots show a very large peak at zero (logarithmic scale) and the distribution is not Gaussian at all. A good rejection function should be bigger than, for example, at least 95% of the found seed energies for single gamma events. To show the rejection value in the previous panels, it could be drawn as a

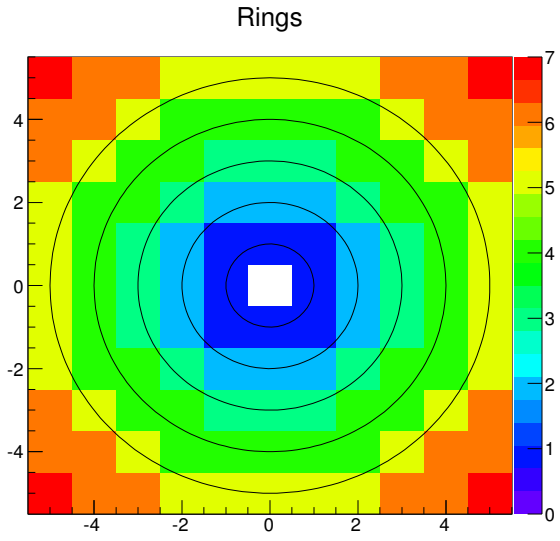

Figure 7: Rings defined in a grid. The numbers on the colour scale represent the ring number. The numbers on the axis represent row and column numbers of the grid.

vertical line (at that value on the horizontal axis) such that 95% of the events should be at the left side of that line.

4.2.4 Quantiles

To apply this idea, different quantiles of the amplitude distributions were calculated. A q-quantile is a value x of the amplitude (or any variable that is being considerd) such that the fraction of entries with an amplitude smaller than x, is equal to q. Graphically, in Figure 9, this means finding the amplitude such that the fraction of the area to the left of this amplitude is equal to q.

Using the *amplitude* distributions to calculate quantiles is not very useful because most of such quantiles have a value of zero especially at rings more than 2 cm away from the origin. This could be seen from the previous figure because the peak at zero was so large that it contained 99% of the events.

Instead, another distribution was considered: for each segment, for each ring, the *maximum seed energy* on that ring was computed. This distribution can be seen in Figure 10. Note that this is no longer any physical quantity both because of the seed energy and because the maximum was taken. It does however provide information for the rejection function. For these *average maximum seed energy distributions* the 90%, 95% and 99% quantiles were calculated. Fits were then done on the resulting profiles and the 95% quantile fit can be found in Figure 11. To clarify, the values of the darkest coloured region, for example, show that 50% of the 10,000 events had their maximum seed energy (for that ring) within this coloured region. The figure also contains a fit on a partial range but this will be discussed later in Section 4.2.6.

For the fine segment the fit result is poor, again due to the values near the origin, similar to the fits on the average maximum. The coloured quantile regions show that the fluctuations in the fine segments are large compared to the fluctuations in the coarse segments, both because the regions are larger and because the 95% quantile is further away from the average. The large fluctuations will make it hard to discriminate between single photons and two π^0 photons.

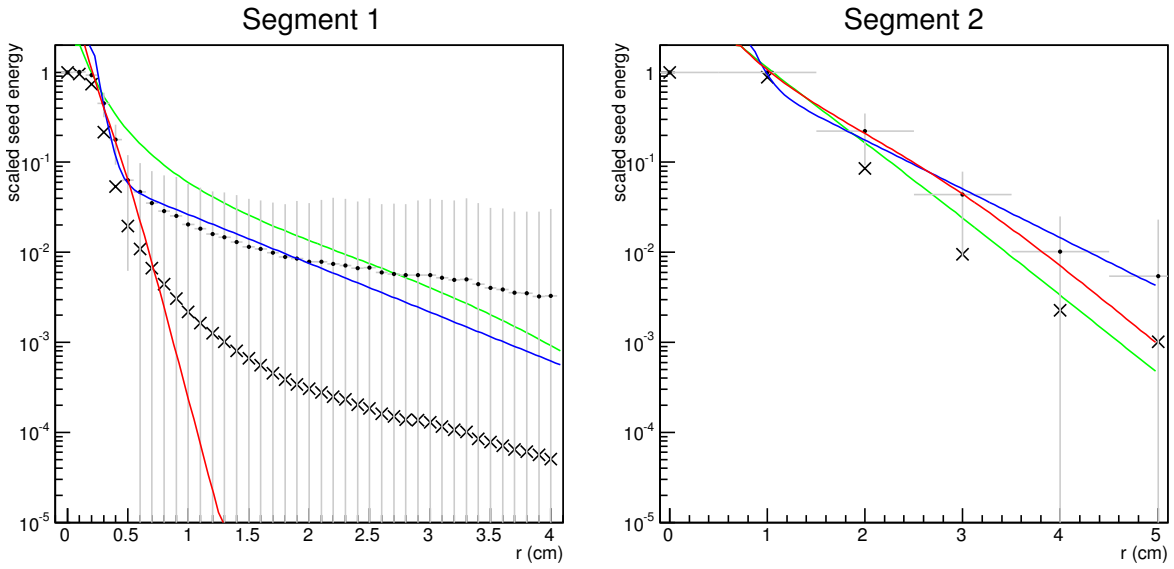

Figure 8: The *average maximum* of 10,000 single gamma events shown as a radial distribution: $\frac{1}{2\pi r}\frac{dE}{dr}$. The dots with error bars show the event average of the *maximum seed energy on a ring* as explained in Section 4.2.3. The crosses show the average (instead of maximum, as in Figure 6) for reference. The blue line is the *double exponential* fit, the red line is the *damped cauchy* fit and the green line is the cauchy weight function (not fitted) *including* the rejection ratio that was being used before this study.

4.2.5 Energy dependence of shower profile

As explained before, the seed energies of every sample are scaled so that the centers coincide. By doing this it is assumed that the shape of the profile is always the same. However one might expect the profile to depend on the energy by more than a scaling of the amplitudes. In Figure 3 an example of an electromagnetic shower profile was shown at different longitudinal depths. It can be seen that the transverse profile is more peaked at lower depths. For the FoCal detector and the particles of interest, the peak in the longitudinal direction should always be located around segment 2. The location of this peak does depend on energy: for high energy particles the peak is at a higher depth. Therefore the shape of the profiles measured in each segment should depend on the energy of the particle. However the longitudinal depth of the peak depends logarithmically on the energy so this effect should not be very large. Note that there could also be other reasons for the shape of the profile to depend on the energy.

To test the energy dependence, the samples were divided in three different energy ranges, and for each range the seed energy distribution was calculated, similar to the figure shown in the previous section. The results can be seen in Figure 12.

The plots show that at the start of the detector (segment 0), the average profile for low energy photons is more peaked, whereas at the end of the detector (segment 5), the high energy profile is more peaked. The difference does not seem very large. The plots also show the maximum seed energy at every distance (averaged over all events) and at segment 3 there is a notable difference for the events with low energy. This suggests that even though the average profile is similar, the size of the fluctuations depend on the energy of the particle.

In all other segments (all but segment 3), both the average and the average maximum do not seem to differ significantly. Therefore, for this research it was assumed that a rejection function that does not depend on the energy is sufficient. It might be an idea for further research to find out whether the results can be improved with an energy dependent rejection function.

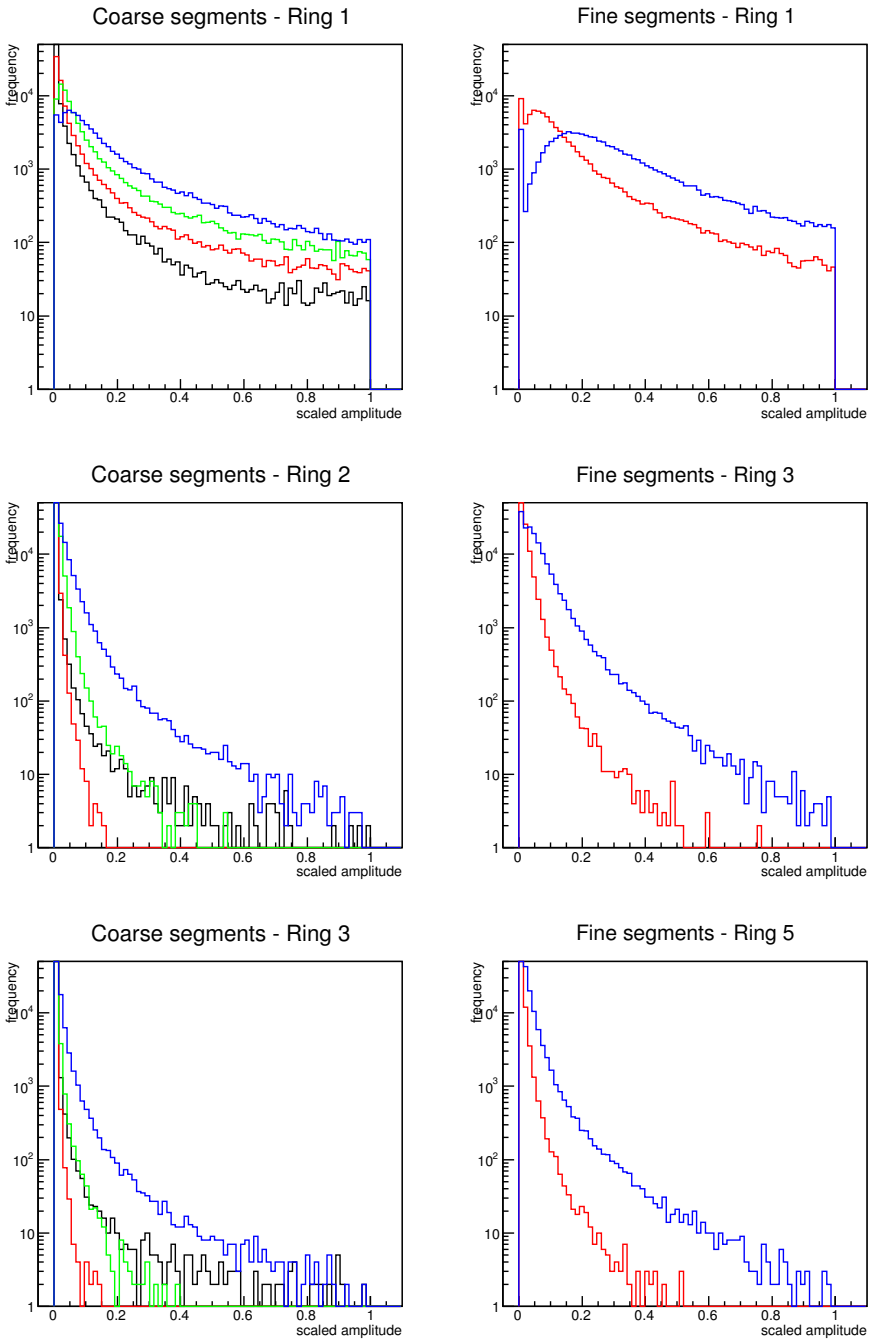

Figure 9: Scaled *amplitude* frequency distribution of the coarse segments (left) and the fine segments (right) on a logarithmic scale. Each panel represents a ring. Each colour represents a segment. The coarse segments 0, 2, 4, 5 correspond to colours black, red, green, blue on the left. The fine segments 1 and 3 correspond to colours red and blue on the right. The amplitudes found in each ring are divided by the amplitude of the center cell. The amount of entries increases at each ring because these rings contain more cells.

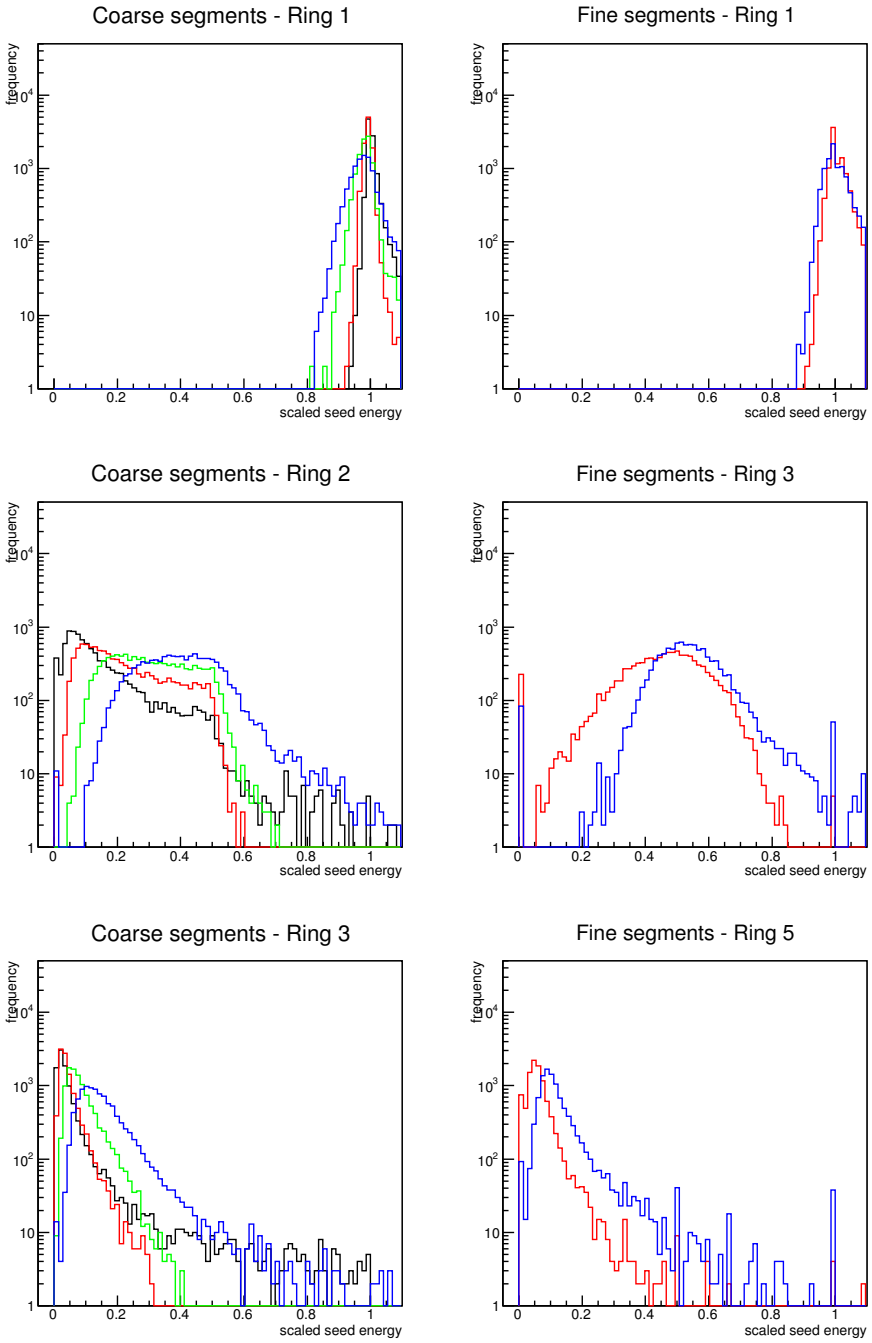

Figure 10: Scaled maximum *seed energy* frequency distribution of the coarse segments (left) and fine segments (right) on a logarithmic scale. Each panel represents a ring. Each colour represents a segment. The coarse segments 0, 2, 4, 5 correspond to colours black, red, green, blue on the left. The fine segments 1 and 3 correspond to colours red and blue on the right. The maximum seed energy found in each ring is divided by the seed energy of the center cell. The distributions at other rings did not reveal any additional insight and were omitted in this figure.

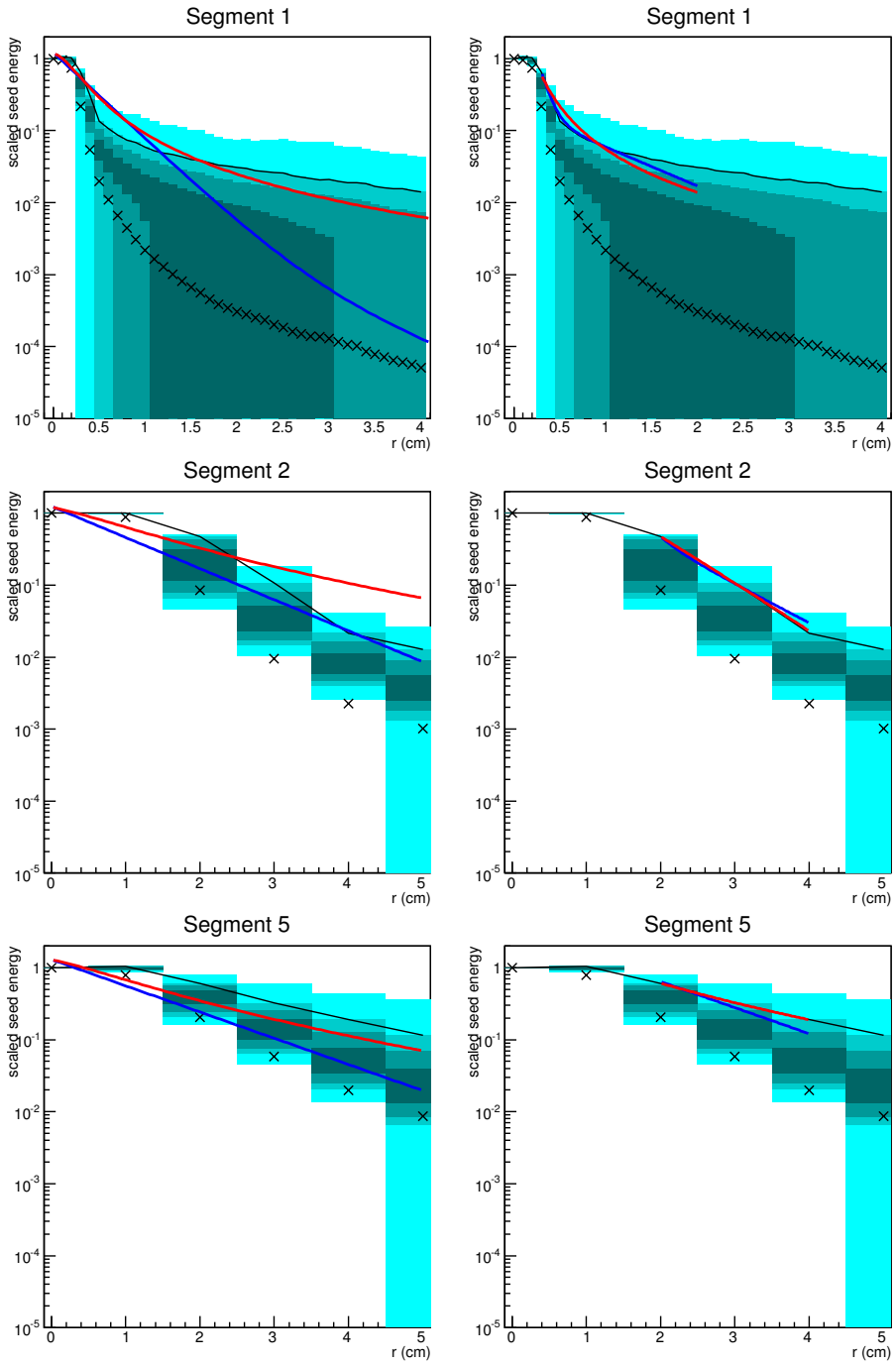

Figure 11: The radial distribution $\frac{1}{2\pi r}\frac{dE}{dr}$ of *maximum seed energies per ring*. The black crosses show the average on each ring. The coloured areas indicate the quantiles of the maximum seed energies: the center area (darkest) shows the middle 50% of all events (25% till 75%). The next area shows the 10% till 90% quantiles, then 5% till 95%, ending with 1% till 99% (lightest). The red line is the *damped cauchy* fit and the blue line is the *double exponential* fit, both fitted on the 95% quantile which is the black line. The left side shows the fits on the full range and the right side shows the fit on a partial range (the other data is equal at both sides).

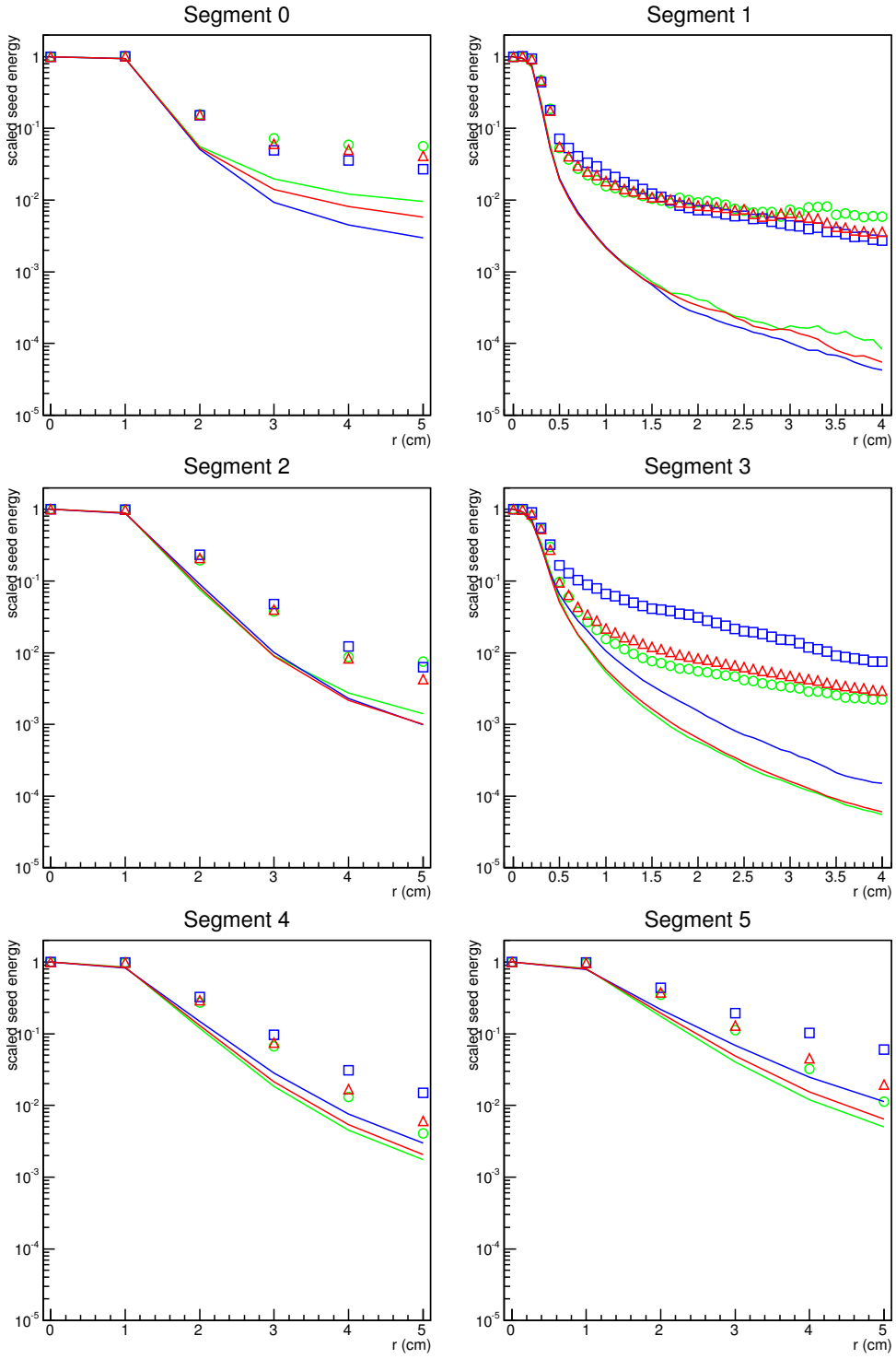

Figure 12: Radial seed energy distribution for different energy ranges. The ranges are as follows: blue is $E \leq 100\,\text{GeV}$, red is $100\,\text{GeV} < E \leq 300\,\text{GeV}$ and green is $300\,\text{GeV} < E$. The lines show the average seed energy. The markers show the event-average of the maximum seed energy at each distance.

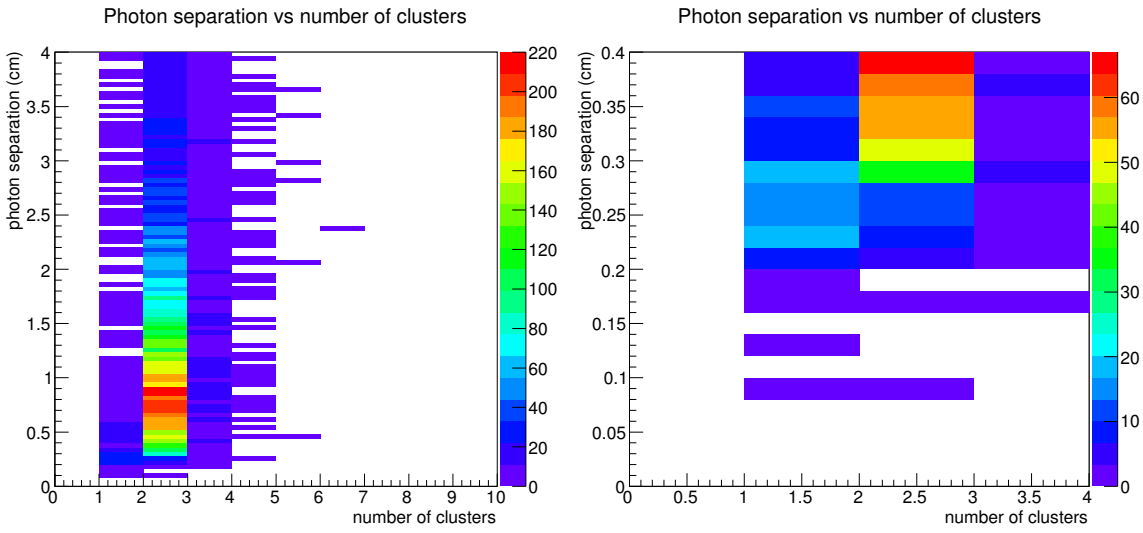

Figure 13: Number of clusters found at 10,000 π^0 events, with the unmodified cluster algorithm, for different distances between the decay photons. The distance is the distance of the incident positions at the detector.

4.2.6 Fit range

In Figure 13 the number of found clusters is shown for different distances between the two incident particles. The graph shows that when the two particles hit the detector less than 0.3 cm apart then they are likely to be seen as one cluster. This behaviour is expected in the sense that the showers almost completely overlap, but it is still important to try to find ways to separate such cases. For larger distances the number of clusters found does not seem to depend on the distance anymore. The fits described in the previous section were therefore repeated but on a different range.

The **MinRing** radius is excluded from the fit because the cells within this radius are always rejected as cluster seeds, no matter what their seed energy is. In previous plots it was also seen that the values near the origin caused poor fit results which is another reason to exclude these values. The fits on the coarse segments were done from 2 cm to 4 cm (rings 2 to 4, inclusive) and the fits on the fine segments were done from 0.3 cm to 2.0 cm (rings 3 to 20 inclusive). The motivation for this was that any potential seeds that were found at larger distances would be rejected by other means (their total energy would be too low) so this rejection function would not have a large effect at higher range.

These fits were done on the 90%, 95% and 99% quantiles of the *average maxima* as described in the previous section. The results of the 95% fit can be found in Figure 11. Comparing the fits to the ones at full range (in the same figure) shows that the functions match the data better, especially in the fine segment. The clusterizer was run with this fit result of the *double exponential* as a rejection function. The efficiency results can be found in Figure 14. The figure shows that the efficiency did not change very much compared to the original data, but the number of clusters that was found did improve slightly, meaning two clusters were found more often. Note that the clusterizer was also run with the 90% and 99% quantile fit results but the resulting efficiencies were not as good. The 90% rejection function was too *low* in the sense that it accepted too many additional seeds resulting in many more events with 3 clusters. The efficiency in this case did become higher. In Section 4.3 it is explained whether this is desired because even though the amount of events with clusters at the incident particle positions is higher (efficiency), these clusters are less accurate because some of their energy is taken by the third cluster.

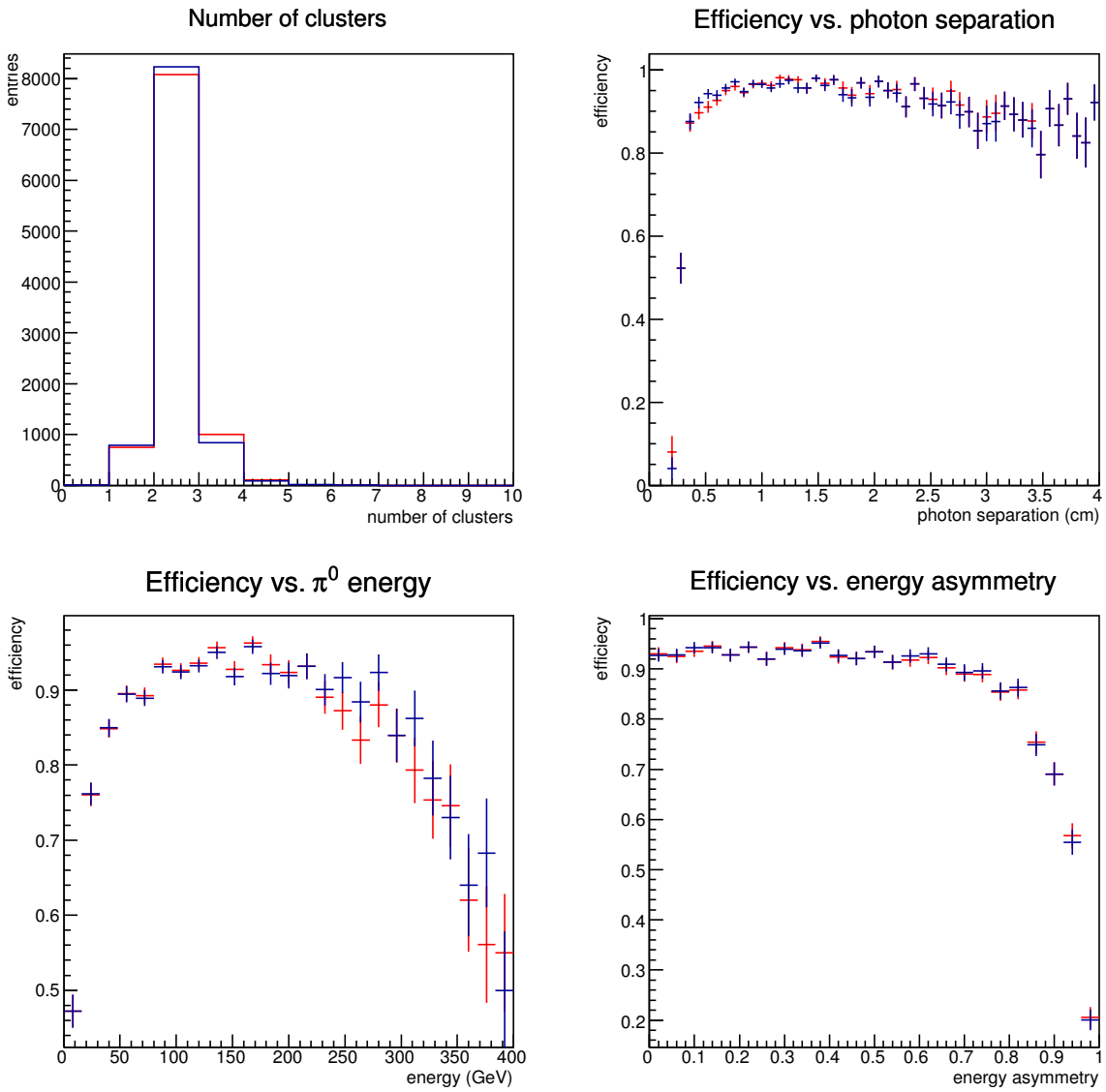

Figure 14: Results of the clustering algorithm for 10,000 π^0 events. The definition of efficiency is explained in Section 4.1. The blue data is the result of the 95% average maximum quantile fit with partial range of the *double exponential* function, as explained in Section 4.2.6. The red lines are is the result of the clustering algorithm with the cauchy rejection function that was used before this research. The average efficiency for the blue data is 85.9% and for the original algorithm it is 85.8%

4.2.7 Ringer distances

In the previous sections the concept of *rings* was used. When the clustering algorithm calculates the expected value at a cell, it uses the *ring number*, multiplied by the pad size of the cell, as a distance value r to calculate the function value. However the (Eucledian) distance between the center cell and a neighbour is not equal for all neighbours on the same ring. The ring number is the nearest integer of the actual distance. For large rings this difference becomes smaller but in the first few rings there is a significant difference.

Figure 7 shows that ring one consists of the 8 cell square. The distance from the center to one of the corner cells is $\sqrt{2} \approx 1.41$ whereas the distance to the non-corner cells is equal to 1. For some more rings these differences are shown in table 2. The table shows that the relative difference between ring number and real distance does tend to zero as the rings get bigger. However, even at ring 20 (which is a distance of 2 cm for a high granularity segment) the distance still differs up to 2%. This might seem insignificant but the rejection functions decrease exponentially. When the function values are calculated on each ring one obtains differences of 20% in values at *all* rings, even the large ones.

To try and improve results, the clustering algorithm was modified to use the real distance (so $\sqrt{x^2 + y^2}$ and not the nearest integer) to each cell when calculating the function value. When using the rejection function that previously gave the best result (shown in the previous section), it turned out that this did not make significant differences in the resulting efficiency.

4.2.8 Mean position

The next modification that was considered was to not use the distance to the origin (of the center cell) but to a mean position. To be more specific, the energy-weighted mean of cell positions is taken over all cells within ***MinRing*** distance to the center cell. The reasoning for this was that mean particle position can differ significantly from the cell center depending on where the particle hits the cell and this is mainly an issue in the low granularity segments. When a particle hits the border of two coarse cells its energy will be distributed evenly amongst them but one of the cells will have a slightly higher amplitude (because of shower fluctuations) and therefore that cell will be used as cluster center. When the cells are 1 cm wide this means the cluster center is up to 0.5 cm away from the particle position which is a significant amount. However, when the mean position is calculated this effect should be greatly reduced.

The clusterizer was executed again with this modification, using the same rejection function as before (shown in the previous section). Since the fit data for the rejection function was based on the *ring number distance* like the original algorithm, this rejection function might not be suitable for the new method that is using real distances and a mean position.

In order to improve the results, new seed energy distributions were computed and instead of calculating the distribution for every ring, each ring was divided into four bins (so they are no longer rings at this point, just distance bins). When creating the distributions, the energy-weighted mean position was calculated and then the distance from each cell to that position was used. Previously, the maximum seed energy on each ring was calculated, and instead the maximum is now calculated for each distance bin. The results of this can be seen in Figure 15.

Due to the higher number of distance bins, the amount of entries for each bin is now lower. Furthermore, as a result of the geometry, when listing distances between squares in a grid, not all distance bins are filled. The first ring, for example, contains cells at a distance of 1 to the origin and cells at a distance $\sqrt{2} = 1.41$, so not all four bins are filled. However, since the distance is taken to a mean position (and not the cell center), it is still possible for these other bins to be filled. The figure shows that this happens at the fine segments, where the mean position can be anywhere in the cell, resulting in a sufficient number of entries for all bins. In the coarse segments, however, the mean position is more often reconstructed at the center of a cell and therefore some bins do not have enough entries. In the figure this effect is seen at segment 0, and also in the other coarse segments to some

Ring number	Minimum distance		Maximum distance	
1	1.00	100%	1.41	141%
2	2.00	100%	2.24	112%
3	2.83	94%	3.16	105%
4	3.61	90%	4.47	112%
5	5.00	100%	5.39	108%
6	5.66	94%	6.40	107%
7	6.71	96%	7.28	104%
8	7.62	95%	8.49	106%
20	19.65	98%	20.4	102%

Table 2: This table corresponds to the rings show in Figure 7. The (Eucledian) distance between the origin and the center of cells on a certain ring is not constant on that ring. This table shows the lowest and highest distance of the cells on a ring.

extent. Even though the fit results are poor in the first two coarse segments, the results in the other segments seem reasonable and so the results were used to run the algorithm.

The clustering algorithm was executed with fit results for the *double exponential* function, using the results obtained from the 90%, 95% and 99% *average maximum* quantiles. It turned out that the optimal results were somewhere between the later two so the procedure was repeated for the 97% quantile. The efficiency results of this can be seen in Figure 16 where they are compared to the results without the modifications of real distances and mean positions. The number of clusters found did increase (when using the modified algorithm) but not by a large amount. The figure shows that the results are better compared to the unmodified algorithm at high energies and low distances (less than 0.5 cm). This is expected because at these distances, the difference between ring numbers and Eucledian distance is larger.

4.3 Too many clusters

In an ideal scenario the algorithm always finds the correct amount of clusters which should be a single cluster for each particle. However it turns out that such a perfect algorithm is not possible because of the fluctuations of occuring in particle showers. In practice, either too few, or too many clusters are found and the goal is to find an optimal balance. This section will explain how it is more favorable to find too many clusters as opposed to finding less clusters than the actual amount of particles.

In Section 1.4 it was explained that one of the goals of FoCal is to measure the energy of particles and then combine clusters to calculate an invariant mass. To do this, the energy of a cluster must be accurate enough. As seen in the efficiency results, the algorithm sometimes finds three clusters when only two are expected. This third cluster will have some non-zero energy which is taken away from the energy of the two valid clusters. If the energy of this third cluster is small enough, the two correct clusters should still have fairly accurate information. This way the invariant mass could still be used to decide whether the particles originated from a π^0 decay.

In Figure 17 the reconstructed energy results of the modified (as explained in Section 4.2.8) algorithm are shown. The histogram shows the reconstructed energy of the two valid clusters for different total numbers of reconstructed clusters. The red line shows the cases with three or more reconstructed clusters, where the other clusters take up some of the energy of the valid clusters. This is shown in the figure as the red distribution is shifted to the left by approximately 0.1 compared to the green distribution, which shows cases with exactly two clusters. The figure also shows the invariant mass distribution which is shown to indicate whether this missing 10% of the energy is significant. The invariant mass plot shows that the red distribution is only shifted to lower values by a very small amount. This means that the third cluster does indeed not take up much energy and therefore the extra cluster not a problem.

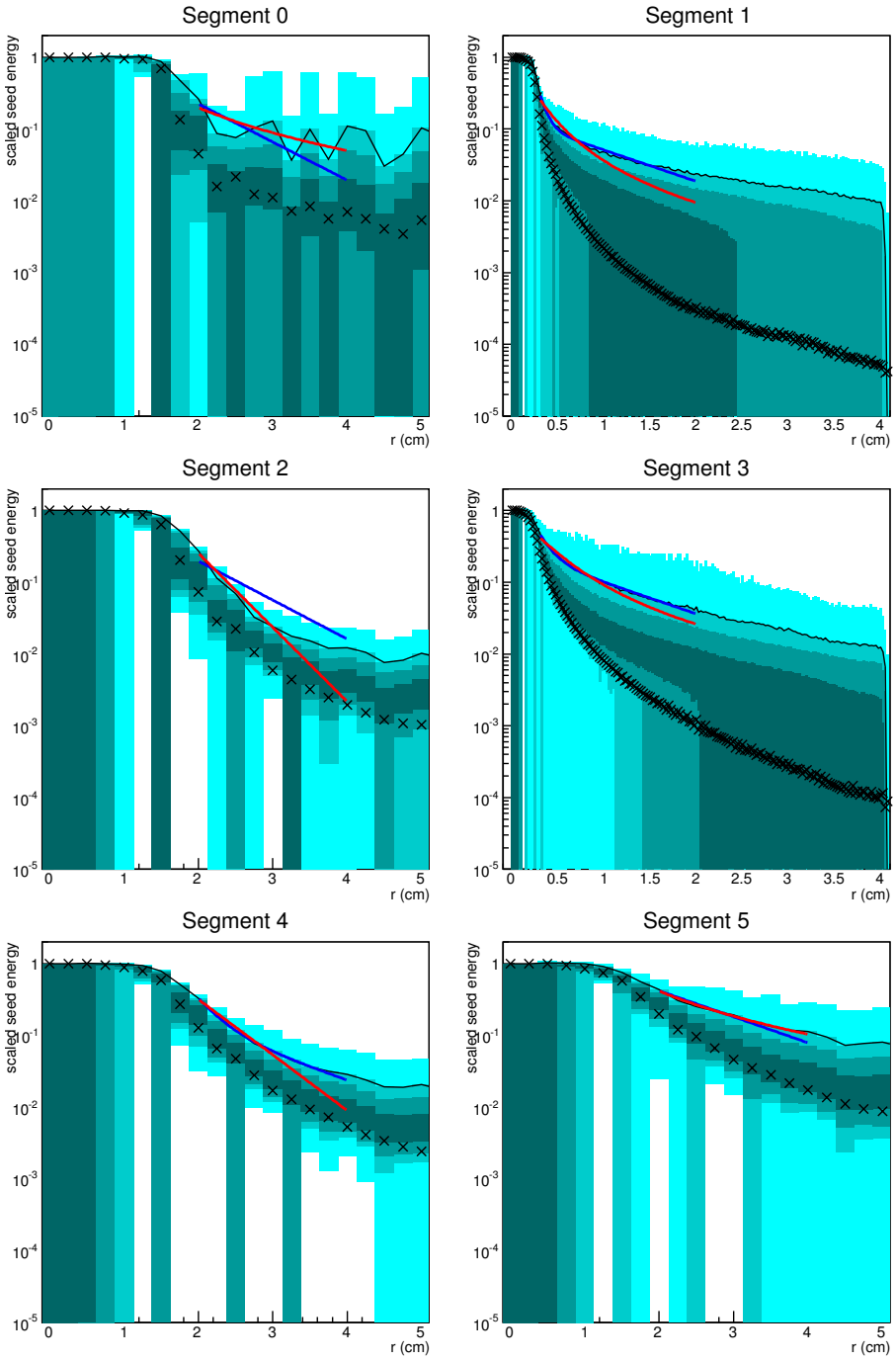

Figure 15: The radial distribution $\frac{1}{2\pi r}\frac{dE}{dr}$ of seed energies. The black crosses show the average on each distance bin. The coloured areas indicate the quantiles of the maxima: the center area (darkest) shows the middle 50% of all events (25% till 75%). The next area shows the 10% till 90% quantiles, then 5% till 95%, ending with 1% till 99%. The red line is the *cauchy* fit and the blue line is the *double exponential* fit, both fitted on the 95% quantile which is the black line. The distance of a cell to the cluster is now the (Eucledian) distance to the energy-weighted mean position.

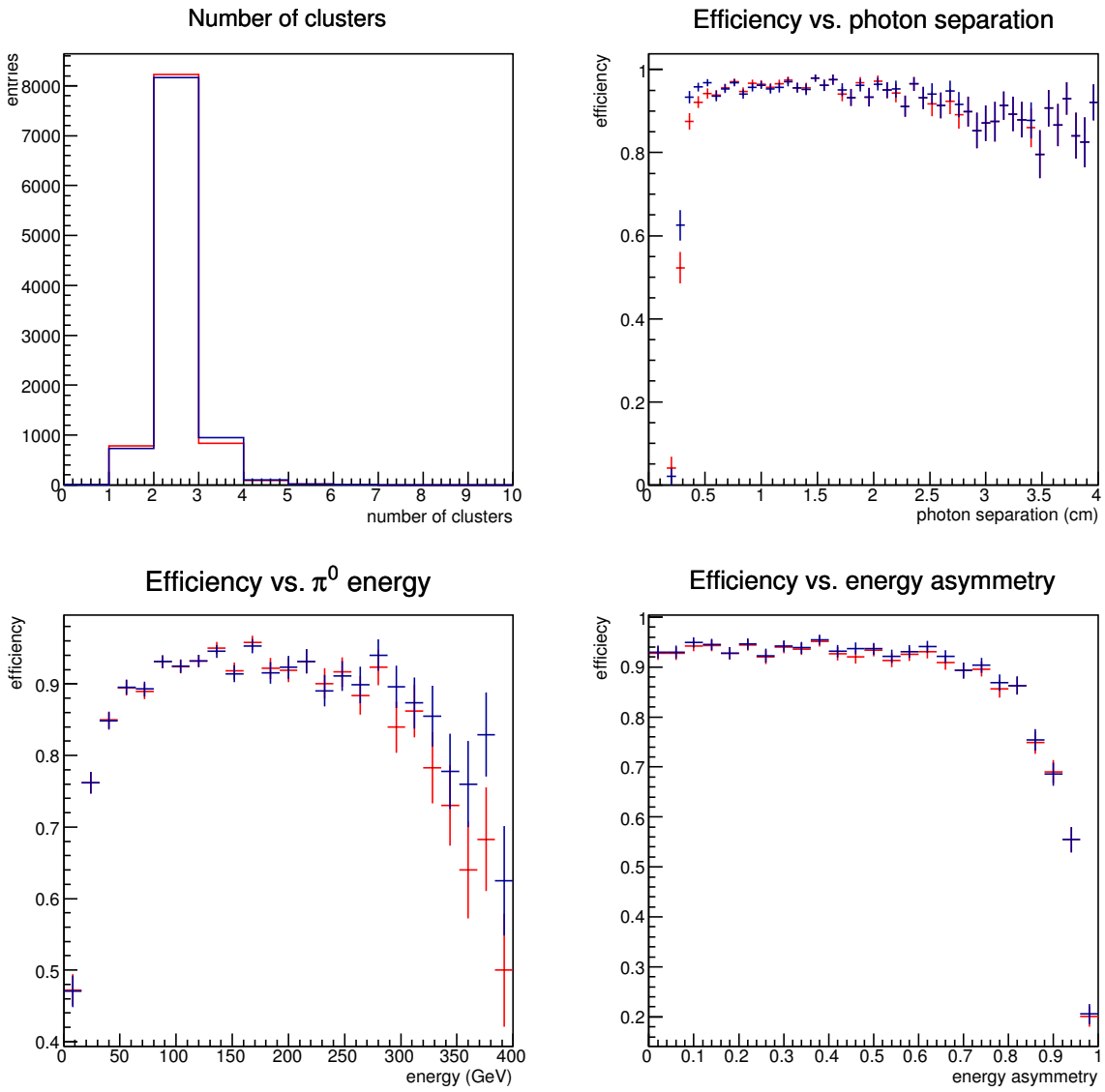

Figure 16: Results of the clustering algorithm for 10,000 π^0 events. The definition of efficiency is explained in Section 4.1. The blue lines are the result of using the *double exponential* rejection function resulting from the fit shown in Figure 15. The modifications to the algorithm explained in Section 4.2.7 were used for this. The red lines are the result of the clustering algorithm without the modifications using the rejection function that was fitted on the 95% quantile shown in earlier sections. The red lines are the same as the blue lines in Figure 14. The average efficiency for the blue lines is 86.3% and for the red lines it is 85.9%

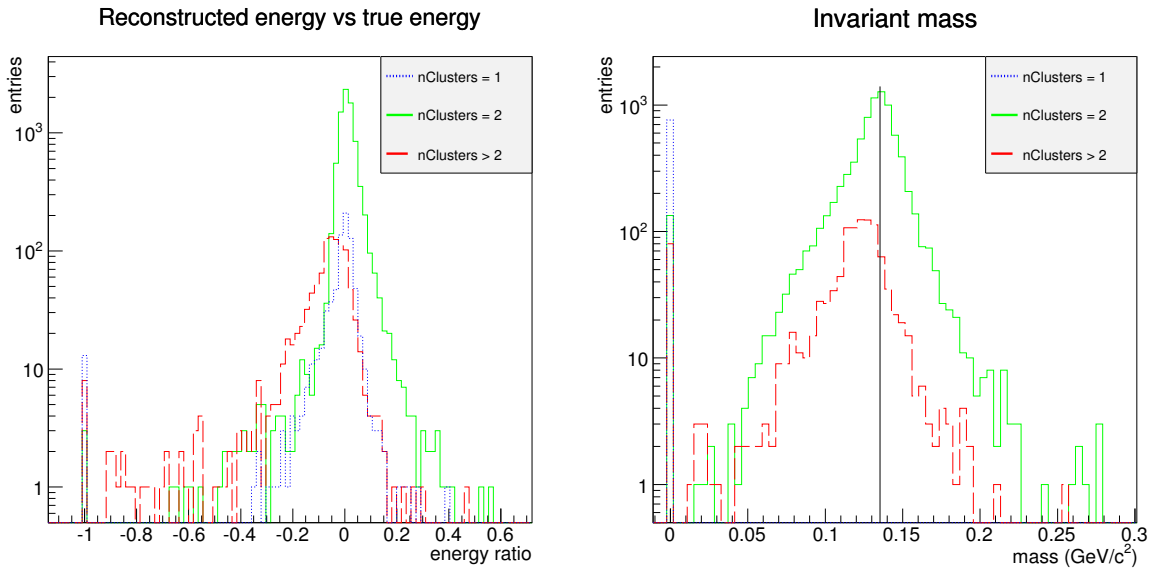

Figure 17: The plot on the left shows the reconstructed energies of the two main clusters as a ratio of the simulated energy of the π^0 particle: $(E_{rec,1} + E_{rec,2} - E_{\pi^0})/E_{\pi^0}$. The plot on the right shows the calculated invariant mass of the two main clusters. The vertical line is the value of the π^0 invariant mass. The different colours indicate the total number of clusters that was found by the algorithm. The main clusters are the two clusters that are the closest to the simulated hit location of the particle. The rejection function that was used for this data was the one explained in Section 4.2.8 (the 97% quantile fit) of which the efficiency results were shown in Figure 16.

In previous sections, the results of the 97% quantile were shown as they were optimal regarding the number of clusters. However, since the extra clusters turn out to not be a problem, the 90% quantile fit was reconsidered. Figure 18 shows the results of the algorithm using this 90% quantile fit together with the previously considered 97% quantile fit results. The plots show a larger efficiency at high energies but as a drawback there are more events with three clusters. To see whether these extra clusters now form a problem, Figure 19 shows the invariant mass distribution. The invariant mass distribution for events with three or more reconstructed clusters is again only shifted left by a small amount. It could be concluded that this is therefore a better rejection function because the efficiency results are better.

4.4 Rejection ratio

The original algorithm used the energy assignment *weight function* multiplied by a *rejection ratio* as a rejection function. During this research these functions were separated and a completely different rejection function was used (with a rejection ratio equal to 1). The purpose of this was to not cause errors in energy assignment when optimizing the rejection function so that any changes in efficiency had to be the result of the changed rejection function.

When comparing the newly found rejection function with the weight function one can see that they are not a constant multiple of each other. The energy *weight function* should match the average transverse energy profile of the particle shower, whereas the rejection function should incorporate fluctuations and it turns out that these fluctuations are not equally large at different distances. In previous sections it was explained that the rejection function was fitted onto the 95% quantile of the maximum seed energies per ring. As a comparison between the weight and rejection function, Figure

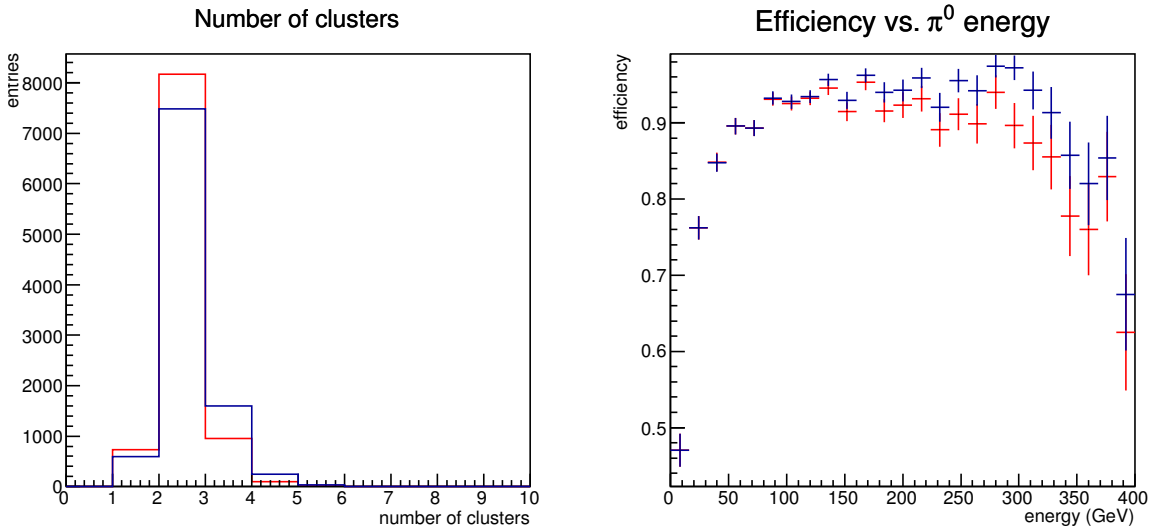

Figure 18: Results of the clustering algorithm for 10,000 π^0 events. The blue lines show the results of the algorithm using the rejection function obtained from the 90% quantile fit whereas the red lines show the results using the rejection function obtained from the 95% quantile fit. The red lines are the same as the blue lines in Figure16. In both cases, the modified algorithm was used. The average efficiency for the blue lines is 87.4% and for the red lines it is 86.3%.

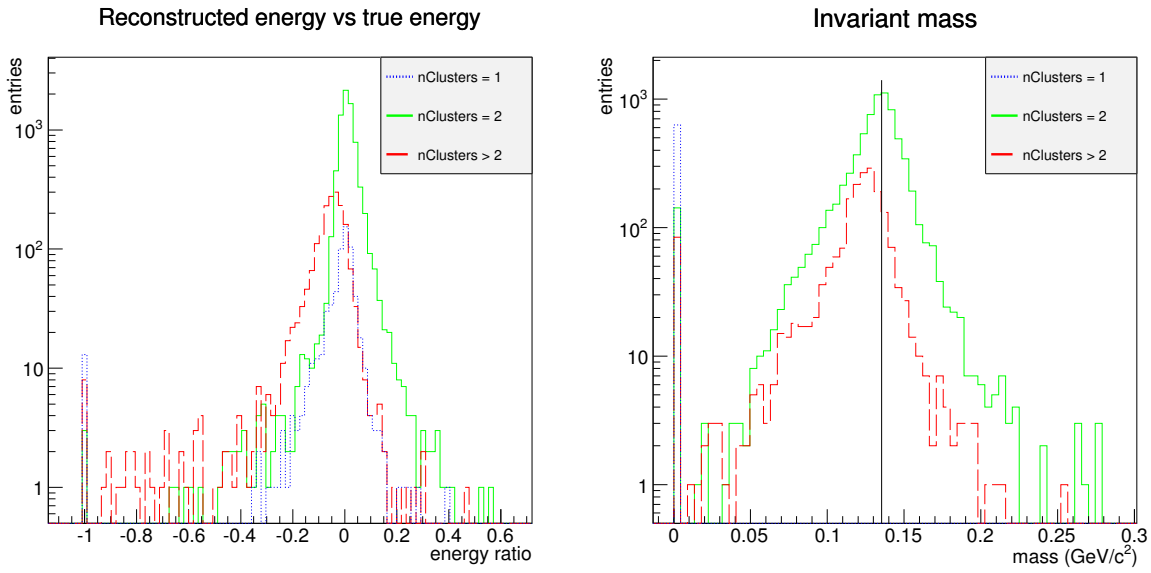

Figure 19: This figure shows the same information as Figure 17 with the difference that the rejection function used here is based on the 90% quantile fit instead of the 97% quantile fit.

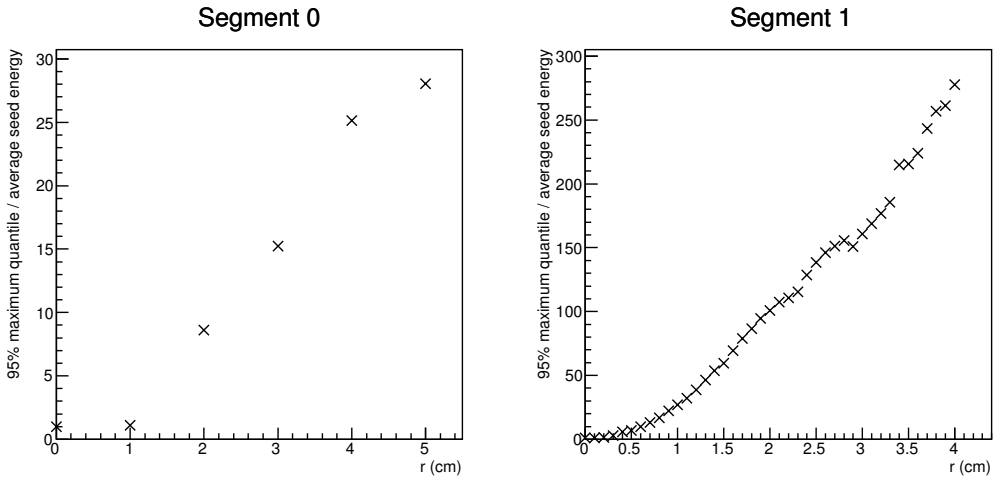

Figure 20: The plots give an indication of what the rejection ratio should be: the data points where the rejection function is based on, divided by the data points that the weight function is based on. In this case the points show the values of the 95% quantile that was fitted in Figure 11, divided by the average seed energy. The other segments showed the same behaviour and were omitted in this figure.

20 shows the 95% quantile data points divided by the average seed energy on each ring. The graphs clearly show that this is not a constant ratio.

It can therefore be concluded that the algorithm should use a separate function for assigning energy and for rejecting cells as seeds.

5 Conclusion and outlook

During this study, the transverse shower profile of gamma particles was investigated in order to improve the clustering of nearby showers. Based on simulated data, the fluctuations in deposited energy were analysed in order to obtain a better rejection function for the algorithm. The double-exponential rejection function has been implemented, as well as a modification regarding the distance of a cell to its cluster center. The results in Figure 14 and Figure 16 show that the new rejection function and the modification to the clustering algorithm yield improved results. There number of events with exactly two reconstructed clusters is higher, and the average efficiency did improve from 85.8% to 86.3%. The distance modification of the algorithm works especially well for distances smaller than 0.5 cm, as can be seen in the second figure. It can also be concluded, as explained in Section 4.4, that the rejection function should be separate from the energy assignment weight function as they are not a multiple of each other.

When considering the number of clusters that the algorithm finds, the current rejection function is near an optimal setting: when the algorithm was run with other parameters either too many clusters were found (when the rejection function is too low) or only a single cluster was found (when the function was too high). When considering the efficiency (as defined in Section 4.1) instead of the number of clusters, a lower rejection function will give better results, as shown in Figure 18. Future research could find the optimal balance between a high efficiency and the correct number of clusters.

Another idea for future research is to improve the weight function that is used for assigning energy to seeds. This function should resemble the transverse energy profile, so the average amplitudes (and not average maxima of seed energies). In Figure 6 on the left, this average was shown together with the weight function in green. For this research the weight function was not modified, but the figure shows, in particular at segment 1, that a better weight function should be possible. A better weight function could result in a better reconstruction of the energies of two clusters, allowing a more accurate invariant mass.

Acknowledgements

First of all I would like to thank my supervisor Dr. M. van Leeuwen for helping me and giving me many good ideas on what to research. Secondly I want to thank my co-supervisor Drs. D. Lodato for helping me understand the essential physics and I also want to thank Prof. dr. T. Peitzmann for his ideas and suggestions. Furthermore I would like to thank Alexandru Babeanu for taking the time to help me understand all the logic of the clustering algorithm and at last I would like to thank Koen Sponselee and Gordian Zomer for their help and opinions.

References

[1] Part of a course *The Physics of Particle Detectors*, given by Prof. H.-C. Schultz-Coulon and Prof. J. Stachel in 2011.
http://www.kip.uni-heidelberg.de/∼coulon/Lectures/Detectors/Free_PDFs/Lecture9.pdf
See also: http://www.kip.uni-heidelberg.de/∼coulon/Lectures/Detectors/

[2] J. Beringer et al. (Particle Data Group), Phys. Rev. D86, 010001 (2012).
http://pdg.lbl.gov/2012/reviews/rpp2012-rev-passage-particles-matter.pdf

[3] The ALICE FoCal collaboration Letter of Intent, A Forward Calorimeter for the ALICE experiment

[4] J. Beringer et al. (Particle Data Group), Phys. Rev. D86, 010001 (2012).
http://pdg.lbl.gov/2012/listings/rpp2012-list-pi-zero.pdf

A Appendix

A.1 Cluster algorithm definitions

A **digit** is the detected signal of a single cell of the segment. All digits have a *flag* that specifies if it can be considered as a **seed** and a **weight** that is used to calculate the fraction of it belonging to a certain cluster. The amplitude of a digit is distributed between different clusters according to this weight.

A.2 Full segment level algorithm

This section describes the steps of the algorithm that is executed separately for each segment of the detector.

1. Initialization: set the above mentioned flag to true for each digit (so the digit is not rejected) and set the weight to 0. Make an empty list of seeds.
2. Sort digits according to amplitude (energy).
3. Calculate **SeedEnergy** for each digit which is defined as the sum of the amplitudes of digits within (and including) the ***MinRing*** radius.
4. In case there are **PreTracks** available these are processed. PreTracks, also called **PreSeeds**, are cluster lists from other segments.
   (a) For each location in the PreTracks list
      i. Find the digit with maximum amplitude within ***MinRing*** distance of the PreTrack location.
      ii. If a digit with non-zero amplitude is found, steps 5-b, 5-c and 5-d are applied for this new seed. This will reject some of the neighbouring digits if those do not have enough energy.
5. Loop over the sorted digits from high to low energy.
   (a) Stop if the amplitude is below ***SeedThreshold***.
   (b) Skip if the flag is set to false, meaning the digit has been rejected as a consequence of part (d)
   (c) Add this digit to the list of seeds.
   (d) Loop over neighbouring digits within the ***MaxRing*** radius:
      i. Add weight to the neighbour digit based on the current digit's SeedEnergy and the ***weight function***.
      ii. If the neighbour is within the ***MinRing*** distance then set its flag to false so that it can no longer be a seed.
      iii. Compare the neighbour's SeedEnergy with an expected value based on the current digit's SeedEnergy, the ***weight function*** and a ***RejectionRatio***. If the neighbour has enough energy to become a seperate seed its flag is unchanged and otherwise it is rejected by setting its flag to false.
      The research done for this thesis focusses on optimizing the expected value mentioned in this step. It is explained in detail in chapter 3.
6. Loop over the list of seeds that has been created from low to high energy. No new seeds can be created, this loop only implements further rejection criteria.

(a) Calculate some properties for the seed. Loop over neighbouring digits within the ***MaxRing*** radius:

i. Calculate weighted number of cells: if a cell has weight contributions from multiple seeds then only the fraction belonging to the current seed is used.

ii. Calculate the total energy, again only taking the fraction belonging to the current seed.

(b) Reject if the weighted number of cells is below ***NCellsThreshold***.

(c) Reject if the total energy is below ***ClusterEnergyThreshold***.

(d) If a seed was rejected then its weight is removed from all neighbours. The seed itself is removed from the seed list.

7. Loop over the remaining seeds to calculate additional properties and create the final cluster list.

(a) The total energy is calculated the same way as in the previous step. This has to be repeated because some seeds may have been removed changing the weights.

(b) The mean position is calculated. The amplitude of a cell is its weight and the mean is taken over digits within 3 rings (inclusive) distance.

$$\vec{r}_{mean} = \frac{\sum E_i \vec{r}_i}{\sum E_i} \qquad \text{sum taken over digits within 3 rings}$$

Other ways of weighing, like using the logarithm of the amplitude, can give better results in some cases but that is not part of this research.

(c) The semi-major and semi-minor width of the cluster are calculated.

This completes the algorithm at the segment level.

A.3 Pre-seeds

The algorithm contains a step that depends on cluster information of other segments. The current clustering algorithm first clusterizes segment 0, 1 and 2. The results of segment 1, the first high granularity segment, are then used as PreSeeds for segment 3, the second high granularity segment. The results of segment 2, a low granularity segment, are used as PreSeeds for segment 4 and 5 which are also low granularity segments.

When this is completed the resulting cluster lists of each segment are combined in order to create clusters at the level of the full detector.

A.4 Full detector level algorithm

At this part of the algorithm the resulting clusters of the previous section, at the segment level, are refered to as sub-clusters. These sub-clusters all have a flag specifying whether they have been **merged** and they have an energy **weight**, which is a variable used for this part of the algorithm and is seperate from the energy of the cluster measured in the first part. The merged flag and energy weight are reset when this part of the algorithm starts.

1. Loop over the segments. First the coarse segments, in this order: 2, 4, 0. Then the fine segments, in this order: 1, 3. The ordering provides some priority to the segments that are processed first. The results of this step are called semi-final clusters. The sub-clusters of the coarse segments are combined into a single list, and the sub-clusters of the fine segments are combined into another list.

(a) Loop over all the sub-clusters in the current segment.

Cluster type	p_0	p_1	p_2	p_3
coarse	$9.375 \cdot 10^{-5}$	$8.532 \cdot 10^{3}$	1.000	$3.898 \cdot 10^{-1}$
fine	$1.605 \cdot 10^{-3}$	$3.968 \cdot 10^{2}$	1.352	$-5.142 \cdot 10^{1}$

Table 3: Energy calibration parameters that are being used in the current implementation of the algorithm (May 2013)

i. Loop over other segments that have the *same granularity*. Find sub-clusters that are not yet **merged**. Note that it is possible that two sub-clusters in another segment are merged with the same cluster of the current segment.

ii. If they are within the ***MinRing*** distance in the (x,y)-plane then mark them as merged:

   A. Energies of the sub-clusters are added. Since the segments are of the same granularity, the values should not need scaling. It is possible to have different weights for each segment but in the current implementation of the algorithm these weights are all set to one.

   B. Position is taken as an energy weighted average of the sub-cluster positions.

iii. When the sub-clusters from all other segments have been searched, the merged clusters create semi-final clusters. The semi-final cluster contains a list of energies, semi-minor, and semi-major widths for each sub-cluster that was merged into it. The energy of the semi-final cluster is calculated according to the following formula:

$$E = p_0 \left(p_1 + \sum E_{sub} \right)^{p_2} + p_3$$

where p_i are parameters that can be calibrated and E_{sub} are the energies of the sub-clusters. The values used in the current implementation are given in table 3. If the resulting energy is not positive then the cluster is not saved. Note that in this case the sub-clusters are still marked as merged so they will not be used again.

2. At this point there is a list of low granularity semi-final clusters and high granularity semi-final clusters. Loop over the high granularity semi-final clusters:

   (a) Find the closest low granularity semi-final cluster that is within ***MaxRing*** distance (where the MaxRing parameter is taken from segment 2, a coarse segment, see table 1). Assign the energy of this high granularity cluster as **weight** to the low granularity semi-final cluster.

   (b) If there is no semi-final cluster within the distance then the high granularity semi-final cluster is thrown away.

3. Loop over the high granularity semi-final clusters again to create the final list of clusters:

   (a) Take the appropriate amount of energy from each low granularity cluster: the energy assigned by the current high granularity cluster divided by the total **weight**. The high granularity energies are only used for this weighting, they are not added to the total energy in another way.

   (b) Use the position of the high granularity semi-final cluster.

4. If there are low granularity semi-final clusters that had no assigned weight yet then they also create final clusters but with no high granularity information. The energy and position information are taken directly from the low granularity semi-final cluster.

This completes the second part of the algorithm.