# Simulation and analysis of clustering for proton CT

Silje Grimstad

Master's thesis in Software Engineering at

Department of Computing, Mathematics and Physics,
Bergen University College

Department of Informatics,
University of Bergen

June 2019

Western Norway University of Applied Sciences

UNIVERSITAS BERGENSIS

# Abstract

A research team in Bergen in collaboration with others is working to develop a proton Computed Tomography (pCT) prototype. It is an image modality that uses proton rather than x-ray to create a CT-image. This can lead to a more precise dose calculation when used in proton therapy.

The proton CT system uses a Digital Tracing Calorimeter (DTC) consisting of multiple layers of ALPIDE chips with the aim of measuring where the protons stops in the calorimeter. The objective of this thesis is to simulate particle hits on a single ALPIDE chip using a realistic data collection of pixel cluster shapes and then use different approaches to reconstruct the clusters. The challenge when reconstructing the clusters is to find a satisfying approach to separate the overlapping clusters. The reconstruction was done by experimenting with various cuts, which had different ways to distinguish abnormal clusters.

The basic method was a template for all the cuts when reconstructing clusters. 3 different cuts were tested: fill percentage cut, asymmetric cut and inactive pixels cut. The different cuts had various parameters to optimise in order to improve the percentage of perfectly reconstructed clusters.

The optimal cut, based on the cuts tested in this thesis, was divided into 3 parts. These include the optimal solution for 0-100 clusters, 100-300 clusters and 300-500 clusters, respectively, in the detector. The inactive pixels cut had the most perfectly reconstructed clusters in the first part. The optimal cut for

the second part, based on the selected parameters, was inactive pixels combined with fill percentage. The third optimal cut, for 300-500 clusters, was the asymmetric cut combined with fill percentage cut. The third optimal cut can have up to approximately 350 clusters in the detector to obtain at least 96% perfectly reconstructed clusters.

# Acknowledgements

I would like to express my gratitude to my supervisors at Western Norway University of Applied Sciences (HVL), Kristin Fanebust Hetland and Håvard Helstrup, and my external supervisor Helge Egil Seime Pettersen. They have been outstanding with their continuous feedback, support, constructive criticism, and guidance throughout this thesis.

Furthermore, I would like to thank the proton CT research team in Bergen for educational meetings, an insight into what they are working on and the pleasure of working with them.

In December I had the opportunity to participate in a beam test in Heidelberg. I want to thank all involved in the experiment, especially the once conducting the beam test.

Finally, I would like to thank my friends and family for providing me with encouragement and support.

Silje Grimstad
Bergen, Juni 2019

# Acronyms

| | |
|---|---|
| **ALICE** | A Large Ion Collider Experiment |
| **ALPIDE** | Alice Pixel Detector |
| **CERN** | the European Organization for Nuclear Research |
| **CT** | Computed Tomography |
| **DTC** | Digital Tracking Calorimeter |
| **HIT** | Heidelberg Ion-Beam Therapy Center |
| **HVL** | Western Norway University of Applied sciences |
| **IMPT** | Intensity Modulated Proton Therapy |
| **MAPS** | Monolithic Active Pixel Sensor |
| **Proton CT or pCT** | Proton Computed Tomography |
| **RSP** | Relative Stopping Power |
| **UiB** | University of Bergen |

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

## 1.1 Background and motivation

A research team with members in Bergen in collaboration with others [1] is working to develop a Proton Computed Tomography (proton CT) prototype based on high energy physics technology. The technology has been developed for A Large Ion Collider Experiment (ALICE) project at the European Organization for Nuclear Research (CERN). The development of a Proton CT system in Bergen is a collaboration between the University of Bergen (UiB), Western Norway University of Applied sciences (HVL) and Haukeland University Hospital.

Proton CT is an imaging modality. It measures the composition of the tissue by transmitting protons through the patient and measuring the energy remaining in the protons when they exit the patient. By measuring how much energy the protons have lost while penetrating the patient, it is possible to create an accurate volume map or an image of the patient. To create the Computed Tomography (CT) image, repeated measurements from different angles will be needed.

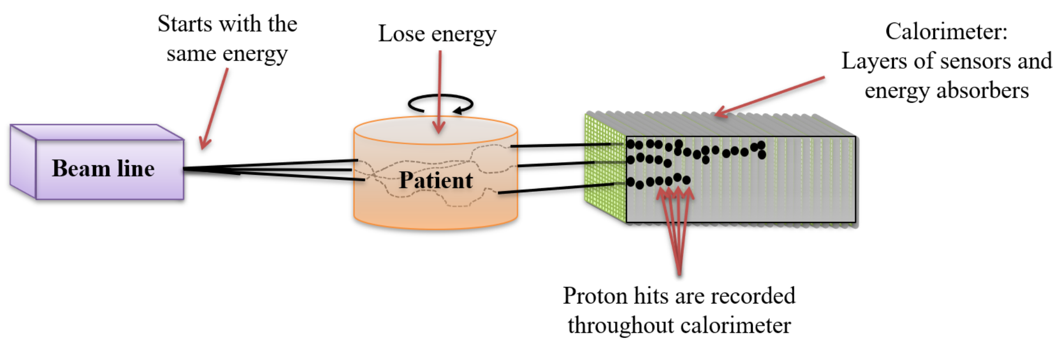



**Figure 1.1:** Proton CT prototype [2].

Figure 1.1 is a schematic setup of the proton CT system from Bergen. It consists of the beam itself, the patient and a calorimeter. The Digital Tracking Calorimeter (DTC) [3] consists of layers of sensors and energy absorbers. The technology behind the sensor is the Alice Pixel Detector (ALPIDE) chip, which is a Monolithic Active Pixel Sensor (MAPS) with a 1-bit digital "hit-or-no-hit" readout [4]. It is a 1024 x 512 pixel grid of 28 x 28 $\mu m^2$ pixels.

The setup shows a beam line that shoots proton particles or heavier ions such as carbon or helium. The particles pass through the patient and into the tracking calorimeter. The tracking calorimeter is divided into different layers of sensors and energy absorbers and the protons will stop in one of them. It is important to have a signal from all the layers such that the track reconstruction can be done.

The particles will start with the same initial energy. Thereafter each proton will lose different amounts of energy in the patient depending on the paths it takes through the patient, determined by the multiple Coulomb scattering. Different types of tissue result in different energy loss, hence the particles will stop in different layers of the tracking calorimeter depending on the energy lost in the patient. Each layer in the calorimeter consists of 9 x (10-12) ALPIDE-chips connected and aluminium absorbers between the planes, with a total area of 27 cm x (15-18 cm). This results in a grid of (1024 x 9) x (512 x 10-12) pixels.

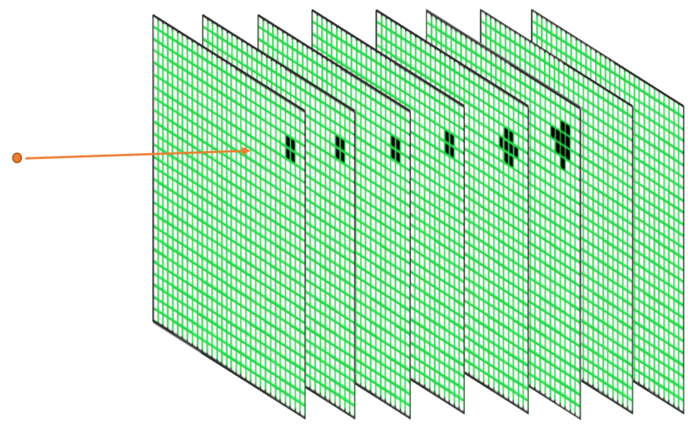


**Figure 1.2:** Demonstration of one particle passing through multiple detectors and stops.

As shown in Figure 1.2, a particle will deposit energy in multiple pixels as it goes through a sensor chip. This is because a proton will eject electron-hole pairs from the Si semiconducting layer of the ALPIDE chips. The many electron-hole pairs will diffuse randomly into nearby pixels, causing neighbouring pixels to activate as well. Pixels where a minimum charge is reached, corresponding to the pre-set threshold value, are activate [5].

Figure 1.2 illustrates a particle passing through layers in the calorimeter. Each layer shows only a section of one detector in order to see how the pixels are activated (dark green pixels) once the particle passes through. The number of pixels activated depends on the energy deposited by the particle. The group of pixels in each layer corresponding to a single particle is called a cluster. The figure shows that the first 4 layers have the same cluster size. Thereafter one can see a rapid increase in the cluster size in the next two layers, followed by no clusters in the last two layers. This is due to the characteristics of a charged particle. The particle will emit a low amount of energy in the first layers, and right before it stops, it will emit a high amount of energy, resulting in a rapid increase in cluster size in the last layers. This is further discussed in subsection 2.1.

Figure 1.2 shows how one particle hit the detectors, but in reality, each detector per readout cycle will have up to ∼500 particle tracks. The readout includes information about which pixel is activated (X and Y positions) in the detector and the time-frame. Therefore, the information must be processed and track reconstruction methods used to find the correct path for each particle.

## 1.2 Problem description

The master thesis will contribute to simulation of the track reconstruction part of the proton CT system, more specifically, to simulation of particles hitting a pixel plane. The contribution involves two parts.

The first part is the creation of a collection of pixel cluster shapes from a Heidelberg experiment. This requires analysing and filtering the data from a test beam experiment and will be further discussed in subsection 3.1.2. This data collection is created in order obtain realistic cluster shapes in the simulation, and in addition, so that it can be used for other simulations for track reconstruction in proton CT as well, as has been done in [6].

The second part includes creating a toy Monte Carlo simulation procedure to randomly pick cluster shapes from the collection and adding this to the detector. This is done with a Gaussian distribution mimicking the data resulting from an Intensity Modulated Proton Therapy (IMPT) scenario. The challenge when reconstructing the clusters is that the more clusters there are in the detector, the more the clusters will overlap and be more difficult to reconstruct. Methods will be needed to distinguish between overlapping clusters and single clusters. Different methods have therefore been developed in order to separate the clusters and thereafter each method is analysed.

## 1.3 Goal and research question

The goal of the master thesis is to create a filtered pixel cluster library and develop and implement different methods to separate the simulated clusters and compare them in terms of particles per pencil beam and efficiency. Particles per pencil beam in this context is defined as how many particles which can hit the detector. Efficiency means how many clusters it is possible to perfectly reconstruct. This means that if 100 particles are shot from the beam into the detector, and the method can reconstruct 96 of them, the efficiency is 96%. If the beam intensity is too high, it will be difficult to separate the clusters from each other and reconstruct them. The number of particle hits per detector in this thesis have been between 0 and 500 because a higher particles per pencil beam will result an efficiency that is too low.

A high efficiency is needed to obtain the best precision and decrease the radiation to the patient when conducting the proton CT. This is further discussed in Chapter 4. The effect of having as many particles per pencil beam as possible, with a acceptable efficiency, is making the duration for taking a proton CT shorter for the patient.

This work aims to answer the research question of which method provide the best result in terms of efficiency compared to particle per pencil beam.

## 1.4 Evaluation and expected result

The final result is the choice of which method will give best results in terms of efficiency and particles per pencil beam. In addition, the second expected result is a filtered pixel cluster shape library. The data collection will be used in other simulations as well, as has been done in [6].

Quantitative methods and case studies using both comparative study and simulations will be used to achieve the expected results.

## 1.5 Report outline

This thesis is divided into the following chapters.

**Chapter 2 - Background** is divided into two subsections and contains a brief background for the relevant domains of this thesis. These are the medical background and the software that was used. The medical background gives a short introduction to cancer, proton therapy, and how proton CT is related to this. The Software subsection gives a presentation of the framework ROOT, and why it was used.

**Chapter 3 - Implementation** is divided into two parts. Firstly, creating a data collection of pixel cluster shapes and secondly simulating particle hits on a single detector and using methods to reconstruct the clusters.

**Chapter 4 - Results and evaluation** contains the results from reconstructing clusters using the different methods described in chapter 3. Additionally, it contains an evaluation of the results.

**Chapter 5 - Conclusion and future work** involves a summary of the results achieved and a proposal for further work based on the findings of this thesis.

# Chapter 2

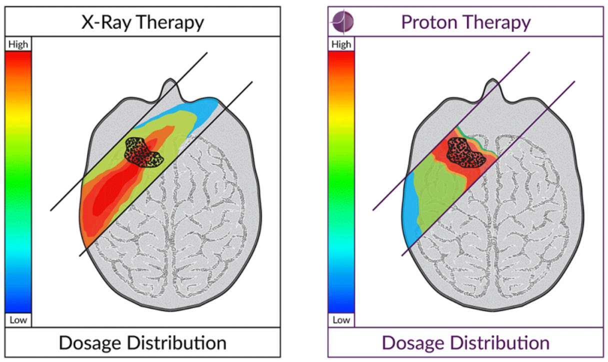
# Background

This chapter contains the theory of the relevant domains of this thesis. It will give a brief introduction to the medical aspect of the thesis and then present the software framework ROOT that was used to develop the project.

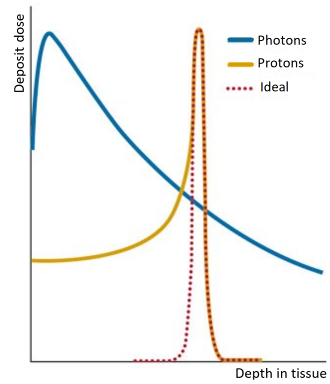
## 2.1 Medical background

Cancer is a common term for the uncontrolled growth and spread of cells [7]. It can occur in every organ or tissue in the human body. There are three main types of treatment for cancer, including surgery, radiation therapy and chemotherapy. Which one to use is determined by several factors such as cancer type, stage, organs at risk adjacent to the tumour and progression [2].

The last decades have been characterised by a significant increase in the number of patients treated with proton therapy, which is a type of radiation therapy. There are 92 facilities worldwide which offer proton therapy and several more are under construction or at a planning stage, including two facilities in Bergen and Oslo [8].




**(a)** Dose distribution in x-ray therapy and proton therapy [9].




**(b)** The Bragg curve [10].

**Figure 2.1:** Illustration of how the protons and photons deposit dose during proton and x-ray therapies.

Proton therapy uses protons to irradiate the diseased tissue as demonstrated in figure 2.1a. The protons are shot through the patient with a high amount of energy and the proton will deposit almost all its energy right before it stops. The goal is therefore to concentrate the majority of the energy from the proton in the diseased tissue of the patient.

The Bragg curve is illustrated in Figure 2.1b showing how much energy protons lose in the tissue and how much photons are absorbed. Protons have high energy and lose minimum energy before the Bragg peak, which results in low damages to the surrounding tissues in front of the tumour. Then all the remaining energy is deposited in the tumour. Photons (used in traditional x-ray), on the other hand, deposits a higher dose to the surrounding tissue before the tumour, and after the tumour as well. Figure 2.1a shows how the radiation is distributed to the surrounding tissue for both x-ray therapy and proton therapy. Proton therapy is preferred for some cancer treatment because of the limited amount of dose to surrounding healthy tissue. However, it requires detailed planning and preparation prior to performing the treatment.

X-ray CT can be used to plan the exact dose calculation for proton therapy treatment. An x-ray CT is an image modality which combines a series of x-rays measurements from different angles, to produce cross-sectional slices [11]. Using the datasets generated by the CT scans, the electron density of the tissue can be calculated [12].

As mentioned in subsection 1.1, a proton CT system in Bergen is in development. Using a proton CT system rather than an x-ray CT it can potentially lead to a more precise dose calculation for the proton therapy treatment. By using proton CT instead of x-ray CT the relative stopping power (RSP) directly from the proton CT scans can be used. RSP is today calculated based on the data from the x-ray CT scans. The RSP defines how much energy the protons will deposit in the patient, which is crucial knowledge in calculating the dose for the proton therapy treatment.

## 2.2 Software

The ROOT framework [13] developed by CERN was used during the project for the master thesis. ROOT is a scientific software framework which is based on the programming language C++. However, there is integration with other languages such as R, Mathematica or Python as well. The framework provides good possibilities for managing statistical analysis, visualisation, storage and large data processing.

The framework uses the Cling C++ interpreter to compile the program. ROOT offers a library of objects such as histograms for visualisation and trees for data storage, in addition to the standard C++. One of the main advantages of ROOT is that it is easy to visualise results in histograms. Histograms may be created by code or different ROOT objects such as TTree may be used to draw histograms.

ROOT is frequently used in the Proton CT research team, and this framework was therefore suggested to develop the master project. This will also make it easier to integrate the solution with other programs that will be developed in the future. In addition it will be easy for the research group to use the program because they know the framework.

# Chapter 3

# Implementation

This chapter will provide a deeper understanding of how the solution was implemented. The implementation was divided into two parts. The first part was creating a representative data collection of pixel cluster shapes which would then be used for further simulation. The second part was simulating particle hits on the detector by using the data collection. Different methods were then used to reconstruct the clusters. The methods were thereafter analysed in terms of efficiency and particles per pencil beam.

## 3.1 Creating a data collection of clusters

A data collection of clusters was created using data from a beam test in Heidelberg [14]. The data was used to obtain a good, representative collection of real pixel cluster shapes.

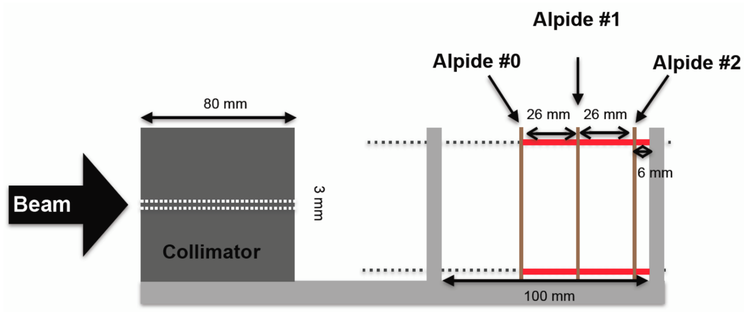
An alternative solution to creating a representative data collection would have been to create a mathematical model which generated a data collection of pixel cluster shapes. However, the experimental data from the beam test were a good, representative and realistic data collection and were therefore used instead of generating a mathematical model.

In order to create the collection, a few pre-processing steps were applied to the data from the beam test. The steps included analysing and filtering the data to obtain a usable collection. The filtering methods removed apparent double clusters, clusters which lasted over multiple readouts, clusters including holes, and separated the remaining double clusters.

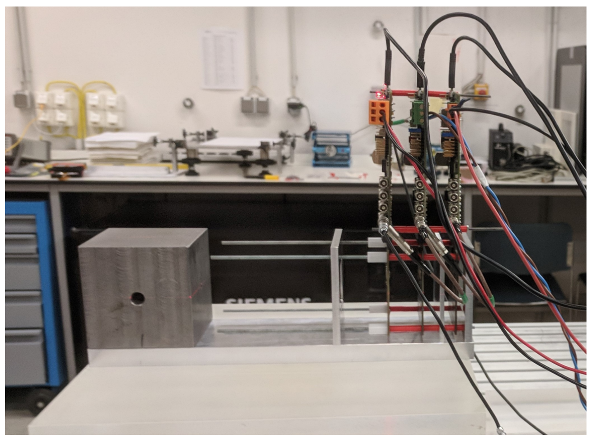
### 3.1.1 Data acquisition

A beam test was conducted at the Heidelberg Ion-Beam Therapy Center (HIT) in July 2018 where helium and proton particles were used [14]. The test was performed with 3 ALPIDE chips in a telescopic layout as shown in Figure 3.1. The purpose of the experiment was to examine the readout efficiency of the chips and view the cluster lifetime, shape, etc. of each particle hitting the detectors. As a result, the system was able to read out 220138 activated pixels during one of the runs. The data for that exact run were written to a single file, which is referred to as the raw data file in this thesis.

The helium particle data were used in the project for the master thesis. The reason why helium particles were used instead of proton particles is that helium has a larger cluster size than protons. It is therefore, more difficult to use the helium particles in the simulation because a larger cluster size results in more overlapping of clusters in the detector, and therefore makes it harder to conduct cluster reconstruction. A more difficult cluster reconstruction was therefore preferred to make sure the reconstruction can be done for even larger cluster sizes than protons. In addition, a model was created which shows the correlation between the deposited energy (measured in keV/µm) and cluster sizes [15]. It is therefore possible to use this data collection for protons-like cluster sizes, if a subset of the cluster size distribution is selected. The Bragg curve for Helium is also very similar to the proton, except that the bragg peak has a even narrower width and needs more energy to reach the same




**(a)**



**(b)**

**Figure 3.1:** Illustration the beam test setup [14].

destination.

Each line in the raw data file corresponded to information about one single activated pixel in a detector with a given `event id`. A section of the data is given in Table 3.1 and illustrated in Figure 3.2. This is the information given from a readout of `plane id` 1 and `event id` 32. The cluster is read out from approximately the center of the chip.

The raw data includes, among other variables, an `index, plane id, hit id, x and y positions, time-stamp, run` and `event id`. The `index` is a counter from 0 to the last readout event in a single `run`, which was up to approximately 220 000 in this dataset. The `plane id` is a value between 0,

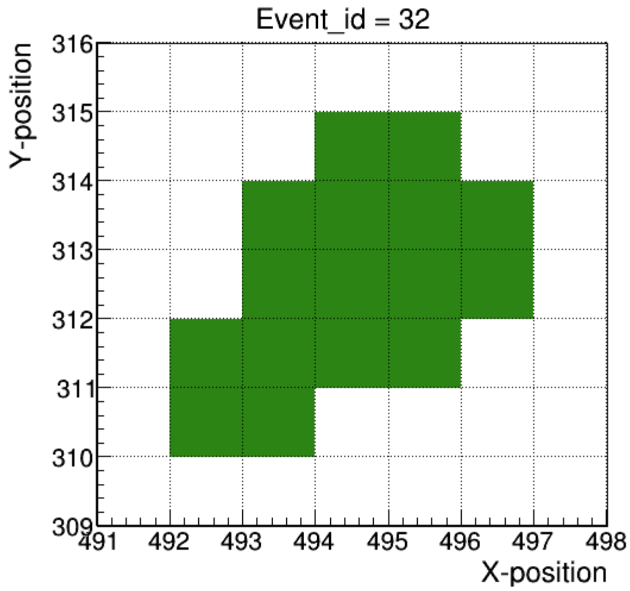



**Figure 3.2:** Cluster with `event id` 32 and `plane id` 1.

| Index | Plane id | Hit id | X pos. | Y pos. | Time-stamp | Run | Event id |
|---|---|---|---|---|---|---|---|
| 0 | 1 | 0 | 492 | 310 | 21 600 | 425 | 32 |
| 1 | 1 | 1 | 493 | 310 | 21 600 | 425 | 32 |
| 2 | 1 | 2 | 493 | 311 | 21 600 | 425 | 32 |
| 3 | 1 | 3 | 492 | 311 | 21 600 | 425 | 32 |
| 4 | 1 | 4 | 493 | 312 | 21 600 | 425 | 32 |
| 5 | 1 | 5 | 493 | 313 | 21 600 | 425 | 32 |
| 6 | 1 | 6 | 495 | 311 | 21 600 | 425 | 32 |
| 7 | 1 | 7 | 494 | 311 | 21 600 | 425 | 32 |
| 8 | 1 | 8 | 494 | 312 | 21 600 | 425 | 32 |
| 9 | 1 | 9 | 495 | 312 | 21 600 | 425 | 32 |
| 10 | 1 | 10 | 495 | 313 | 21 600 | 425 | 32 |
| 11 | 1 | 11 | 494 | 313 | 21 600 | 425 | 32 |
| 12 | 1 | 12 | 494 | 314 | 21 600 | 425 | 32 |
| 13 | 1 | 13 | 495 | 314 | 21 600 | 425 | 32 |
| 14 | 1 | 14 | 496 | 312 | 21 600 | 425 | 32 |
| 15 | 1 | 15 | 496 | 313 | 21 600 | 425 | 32 |

**Table 3.1:** Raw data for one particle hit in one plane.

1 or 2 depending on the plane the data was read out from. The `hit id` is a counter of activated pixels for a single event on a plane. Each activated pixel on a single plane at a given `time-stamp`, has an `hit id`, starting at 0 and counting up towards the number of activated pixel in that plane. However, the `hit id` is not serialised throughout the entire `run`, it is just unique in that exact readout from one plane. This is to enumerate the pixel in the cluster. The `x` and `y position` identify the position of the activated pixels in the chip, with a maximum of 511 for `y position` and 1023 for `x position`. `Time-stamp` is the time when the chip was read out and are in units of 25 ns, and the `run` is a number describing that exact run in order to distinguish the different runs. The `event id` is similar to the `time-stamp`, it is a value for one readout, and can be the same for multiple planes. It is not a unique number for one single cluster or plane.

The system was configured such that it read out data from the chip every 20 µs in a given period of time if the chip contained any data. For each time it read out from the 3 planes, an `event id` is given. The `event id` starts at 0 and increased by one for every readout. The frequency was lower than the readout speed in order to capture each cluster. A cluster could "live" over multiple events in the chip. If a readout did not contain any data, it incremented the `event id` and the `time-stamp`. Therefore, `event id`s will not have sequential numbers and and it will vary which plane containing data. Therefore, one will not get sequential numbers of `event id`'s and it will vary which plane containing data.

The data from `event id` 32 on `plane id` 1, as shown in Table 3.1, is one single cluster. However, the readout of one event on one plane does not always include a single cluster. Therefore, filtering and analysing steps were performed in order to obtain a representative data collection of single clusters, which will be further discussed in the next subsection.

### 3.1.2 How the database of clusters is created

The raw data from the beam test in Heidelberg included double clusters in single events, clusters with holes in them and clusters which had a lifetime in same plane over two events, etc. Therefore, a couple of filtering and analysing steps were conducted.

#### Removal of some apparent double clusters

A simple method was used in order to filter out apparent double clusters, illustrated in Figure 3.3. The method consists of looping through all the data and for each event in a single plane, the size, height and width of the clusters are checked. The information on each event in a single plane was treated as one cluster in this method, in order to remove double clusters. The double cluster will usually have a higher value in height, width or size, than a single cluster. This is because the height in every cluster is defined as:

*Height = Maximum y pixel - Minimum y pixel*

The width was calculated the same way but using x pixel values instead. The size of the cluster is defined as how many pixels there are in the data, such as 16 in Table 3.1.

By looking at the distribution of the size, width and height in Figure 3.4, a value to separate on can be determined. That is to say, the value which represents the barrier for abnormal clusters. The distribution of cluster sizes shows that the tail of the graph starts around 25-30. The cluster size of 30 was therefore chosen to be the barrier for the size of a single cluster. Figure 3.4 shows that height and width of the cluster was quite similar. The barrier for those two values are therefore equal and defined as the value 10. If either the height, width or size was greater than the barrier values, the cluster was discarded. The double cluster method removed 1058 elements from the dataset.

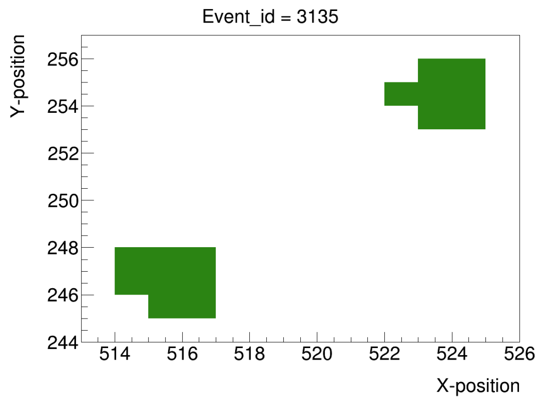



**Figure 3.3:** Double cluster on `plane id` 1 and `event id` 3135.

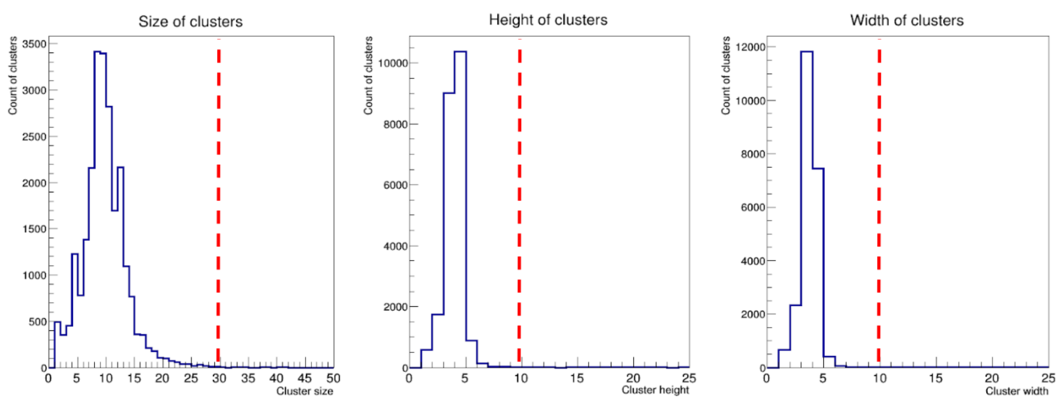



**Figure 3.4:** Distribution of size, height and width of clusters in the raw dataset.

As a result, there were 22 773 remaining clusters to work with within the next method.

### Combining clusters

Another challenge with all the datasets from the ALPIDE chip was that some of the clusters last over two events. Multiple examples are given in Figure 3.5. For instance, `event id` 56 and 57 (top two clusters to the right) have approximately the same x and y pixels activated, with only one pixel less in the second cluster.

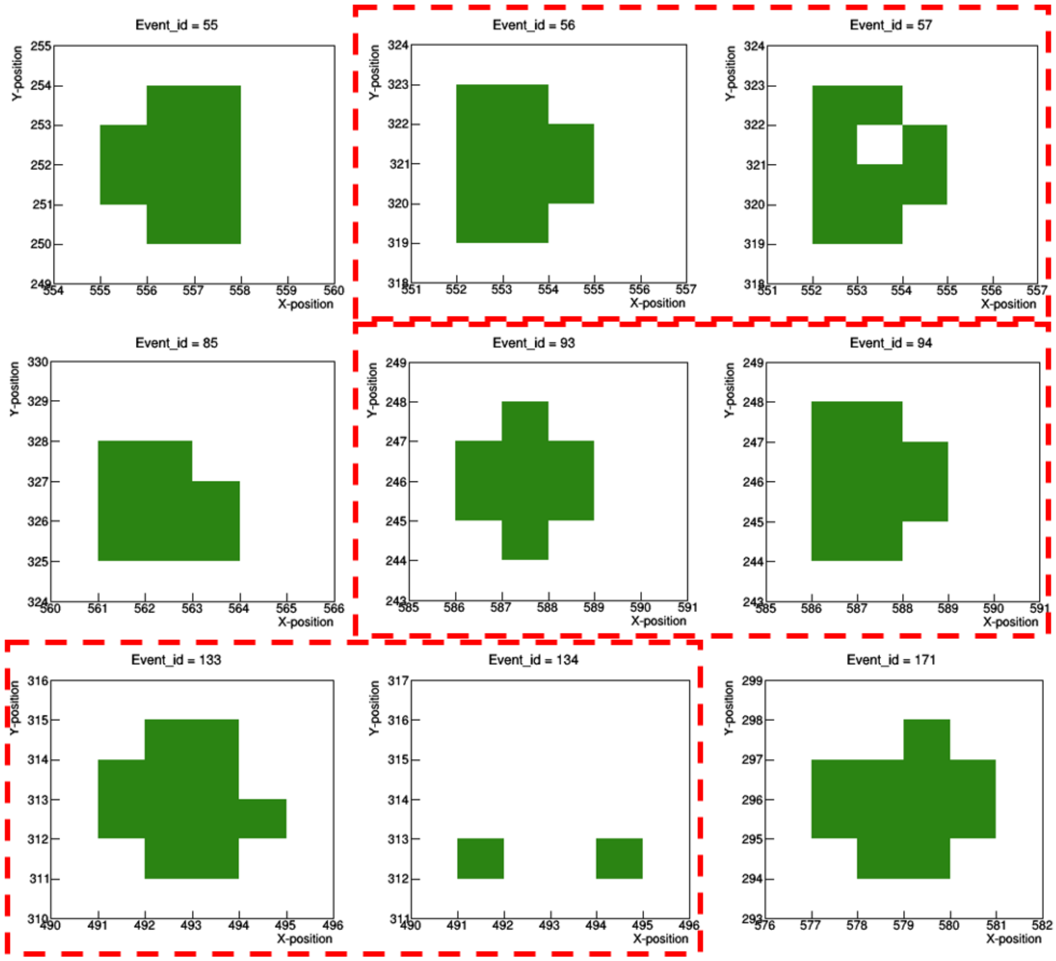



**Figure 3.5:** The 9 first clusters on plane 0. The 3 red dashed rectangles shows which clusters where combined.

The solution was to compare each cluster to the subsequent cluster. If the cluster's event id only differs by one and they were on the same plane, then the two clusters probably originated from he same track. To determine if they should be merged, the larger cluster was compared with the smaller cluster. If the pixels in the small cluster were a subset of the large cluster, excepting a given number of additional pixels in the smaller one, then the clusters are merged. The exception was given in order to allow a slight change in the cluster shape from the two different readouts. The difference was set to 2 in this implementation but could easily have been adjusted if needed for another dataset. The value was chosen after a quick analysis in which all of the smaller clusters had no more than two pixels that were not part of the bigger cluster.

The analysis was conducted to find similarities in a range of 1500 clusters. It was conducted to understand how long the "lifetime" of a particle lasted in the detector and how the shape changed. The conclusion from the analysis was that all the clusters lasted a maximum of 2 readout frames in the readout from the detector. This is consistent with the readout settings which were explained earlier, since the readout period (strobe window) is 20 µs. Additionally, the two clusters which appeared to be the same were compared. All the compared clusters had one smaller cluster which was a subset of the larger cluster, with an exception of 2 pixels. As a result, it was decided to combine all clusters with subsequent `event ids`, where one of the clusters was a subset of the other with an exception of 2 pixels.

### Removed clusters with holes

Another method for filtering unwanted clusters is the removal of clusters with holes thus removing some additional double clusters. A couple of examples of clusters with holes are shown in Figure 3.6.

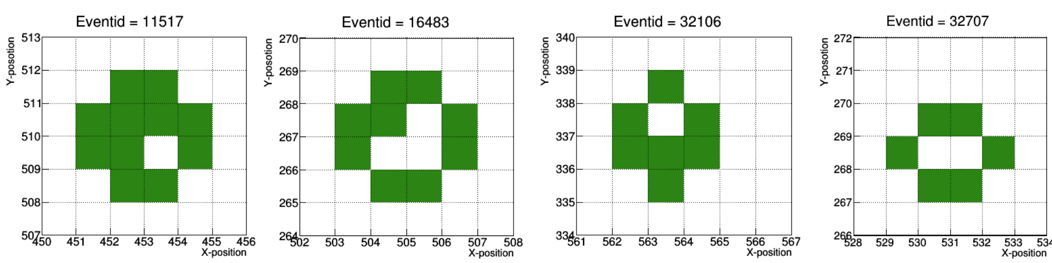
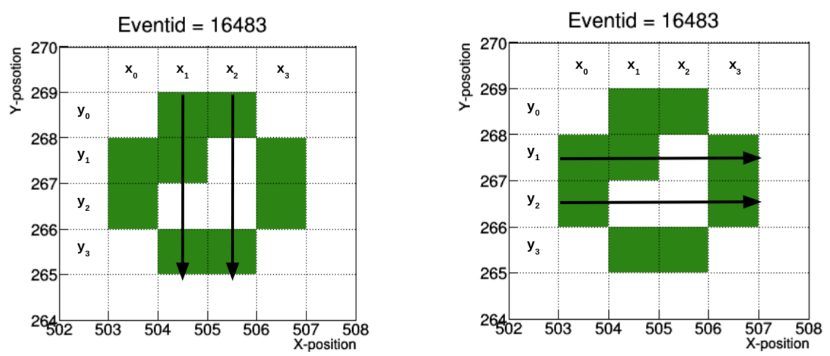



**Figure 3.6:** Examples of hole in clusters. The active pixel is coloured green and the white pixels are inactive.

Figure 3.6 shows 4 examples of clusters with holes. A hole is defined as one or multiple inactive pixels between two active pixels, either in the x or y direction.

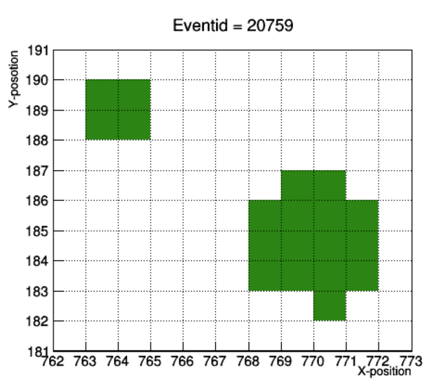



**(a)** Removing holes in x direction. **(b)** Removing holes in y direction.

**Figure 3.7:** Illustration of method for removing holes.

The exception is if the inactive pixel is on one of the outer edges of the cluster. Then the cluster does not include the inactive pixel as a hole. The method is illustrated in Figure 3.7.

In addition to remove holes, the method will simultaneously remove some remaining double clusters. Each cluster is stored in a binary list as a bit pattern, which means that the cluster in Figure 3.8a, will be stored as the list to the right in 3.8b.




**(a)** A double cluster in the detector.

| $2^8$ | $2^7$ | $2^6$ | $2^5$ | $2^4$ | $2^3$ | $2^2$ | $2^1$ | $2^0$ | |
|---|---|---|---|---|---|---|---|---|---|
| 256 | 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 | **384** |
| 256 | 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 | **384** |
| 256 | 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 | **0** |
| 256 | 128 | 64 | 32 | 16 | 8 | **4** | **2** | 1 | **6** |
| 256 | 128 | 64 | 32 | 16 | **8** | **4** | **2** | **1** | **15** |
| 256 | 128 | 64 | 32 | 16 | **8** | **4** | **2** | **1** | **15** |
| 256 | 128 | 64 | 32 | 16 | **8** | **4** | **2** | **1** | **15** |
| 256 | 128 | 64 | 32 | 16 | 8 | 4 | **2** | 1 | **2** |

**(b)** Illustration of how the cluster is calculated to a binary array containing only a list of numbers. The dimensions of the array are the same as the height and the width of the double cluster, which is 8x9.

**Figure 3.8:** Illustration of how a cluster is stored in a binary array.

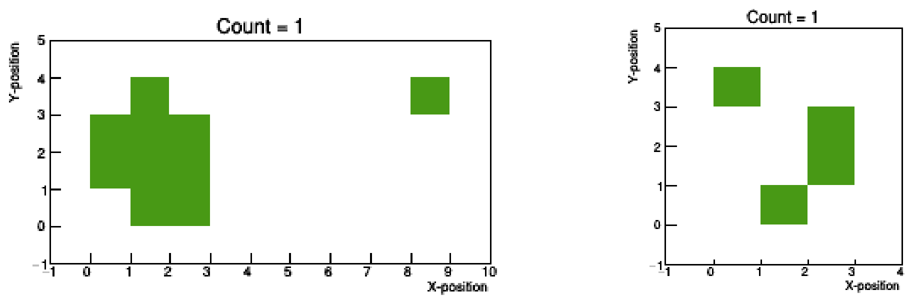
The cluster is placed in a 2-dimensional array with the same height and width as the cluster. Each column represents a value in the base 2 system. These values are calculated by the index, such that index 0 has the value $2^0 = 1$. The index starts at the end of the array and increases moving to the left as seen in Figure 3.8b. To calculate the value for a row, the pixel position on each column is added to the value of the column. An example is how the value 384 is calculated.The index of activated pixels in that row, which is at index 7 and 8 is determined. Then the values from these two columns are added together, $2^7 + 2^8 = 384$. This procedure is done for every row in the array and results in the list [384, 384, 0, 6, 15, 15, 15, 2].

If the list contains the number 0, it means that a double cluster exists, since there are no pixels in that row. Figure 3.8a is a good example of this. Therefore, are all binary lists containing 0, are removed from the data collection.

### The last pre-processing step

A last pre-processing step was conducted in order to obtain a good, representative collection of clusters without errors. A single cluster is here defined as a group of active pixels that are next to each other in the x and y directions, not in a diagonal direction. There were 20 examples in the collection which did not fulfil this definition, and two examples are illustrated in Figure 3.9.

One can clearly see that these two examples contain multiple clusters. However, none of the previous filtering methods was able to remove them. Therefore, a method for separating the remaining double clusters and separately adding them to the collection was developed. The method is also used to separate clusters in simulation in subsection 3.2 and will therefore discuss this in detail there.




**(a)** A cluster which is separated into two stored clusters.

**(b)** A cluster which is separated into three stored clusters.

**Figure 3.9:** Illustration of two clusters which does not fulfil the definition of a single cluster. The "count = 1" means that only one element of each exists in the data collection.

As a result of the method, Figure 3.9a is added as two clusters in the collection, and Figure 3.9b is added as three clusters. 20 wrongly defined clusters resulted in 47 clusters in the data collection.

Consequently, all clusters containing only a single pixel were removed. These pixels were most likely noise in the chip. There were approximately 500 single pixels.

### 3.1.3 Final collection of cluster shapes

The final cluster collection contains approximately 13500 clusters. Each cluster in the collection contains information about mean x and y values, size, positions of the cluster, height, width, cluster shape, event id and if the cluster was combined.

The collection has the following distribution of size, height and width as shown in Figure 3.10. The size has a mean value of 9.7 pixels, the height has a mean value of 3.5 pixels and the width has a mean value of 3.3 pixels.

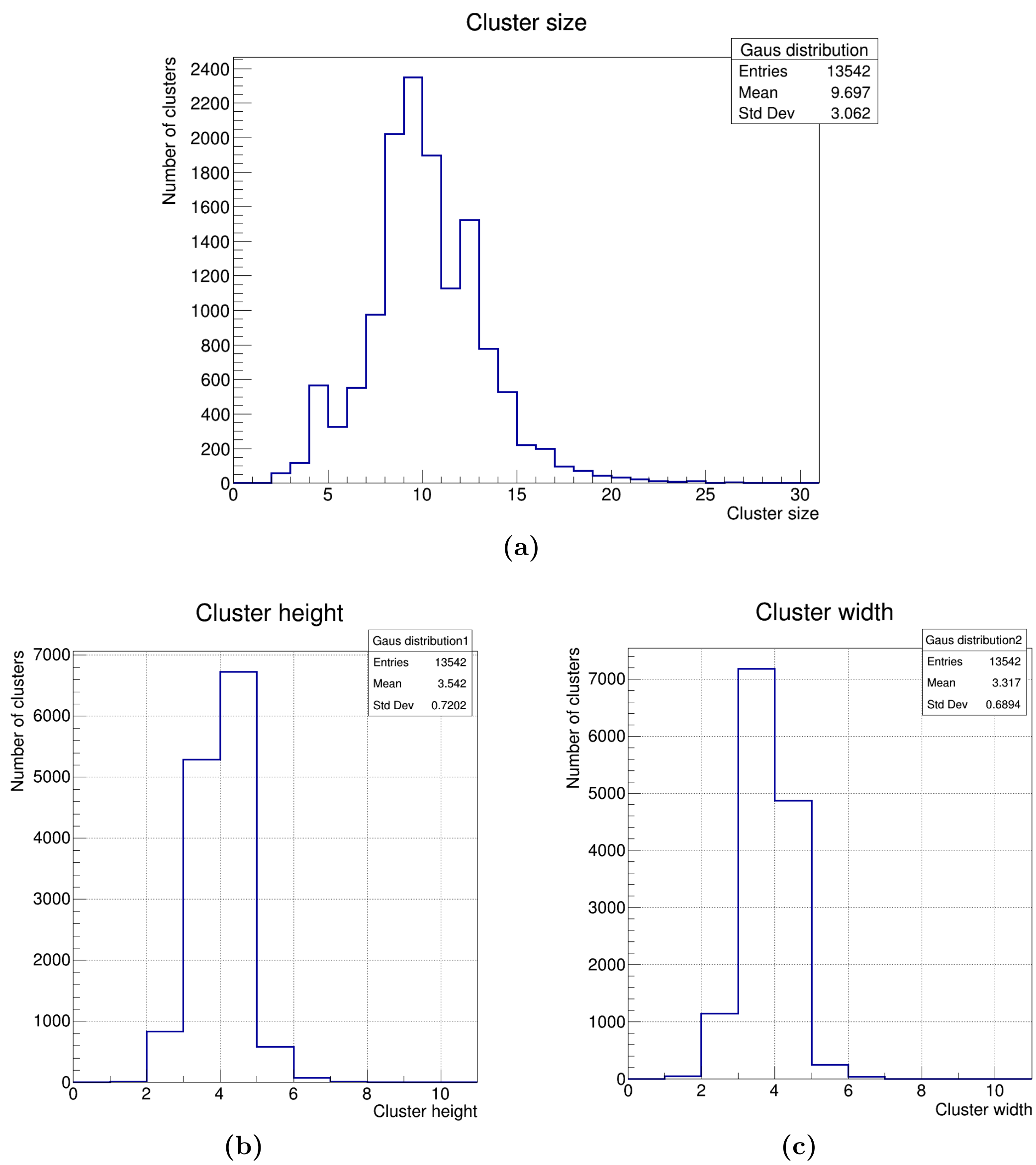



**Figure 3.10:** Distribution of size, height and width.

Additionally, the collection contains around 550 distinct cluster shapes. The most frequent 16 cluster shapes are shown in Figure 3.11. The clusters here have a count between 235 and 1093.

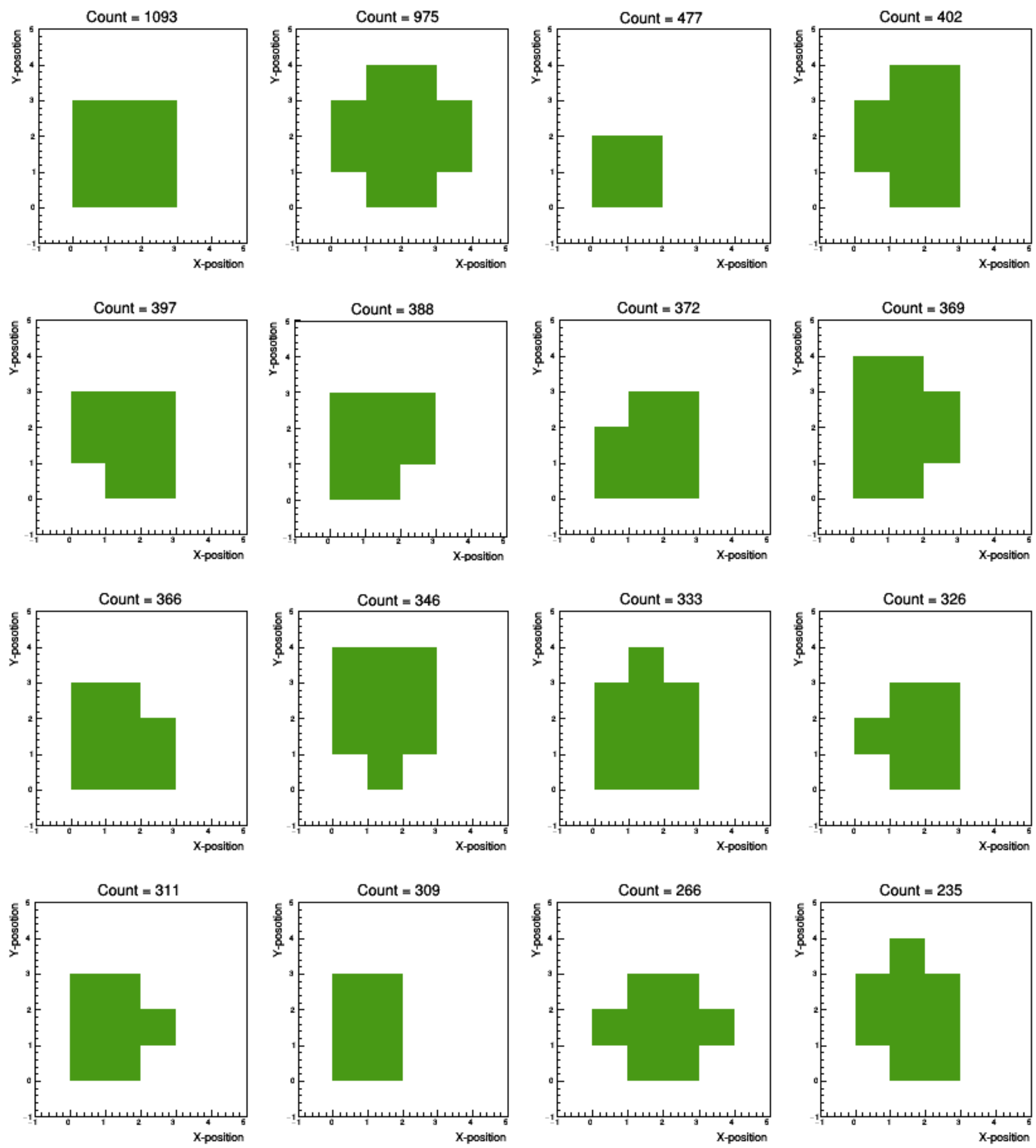



**Figure 3.11:** 16 most frequent clusters in the data collection.

## 3.2 Simulation

Simulation is used in order to imitate a function of a system or process in the real world by means of the use of another system. It is an important tool used in scientific research to mimic the behaviour of a system. Monte Carlo simulation is a type of simulation where one uses a computer program to represent a system based on a mathematical description, or model. The simulation also makes it possible to test each component of a system separately and simulate a system which does not yet exist [16].

In this project, simulation is used to study the pixel cluster reconstruction part of the proton CT system. The "toy Monte Carlo" simulated system is used to imitate a beam shooting helium particles at a given beam size towards an idealised pixel detector. Thereafter, different methods are used to reconstruct as many clusters in the chip as possible. Finally, an analysis is performed to evaluate how many correct clusters the algorithm was able to find.

### 3.2.1 Method for simulation

The method for simulation consists of generating particle hits on the ALPIDE chip. In order to conduct the simulation, parameters such as the lateral beam size and the count of particles hitting the chip are selected. Thereafter, random x and y points from a Gaussian distribution are selected as shown in Figure 3.12. The mean values for the x and y positions are 512 and 256, respectively. Both Figure 3.12 and Figure 3.13 show examples of a simulation with 1000 hits and a beam size corresponding to a circular Gaussian with a standard deviation of 4 mm, or approximately 143 pixels.

Figure 3.12 shows the x distribution on the left, and y distribution on the right. The standard deviation for the two figures is slightly different. The left graph has a standard deviation of 146.8 pixels, and the right graph have a value of 119.5 pixels. This is because the simulated ALPIDE chip has a dimension of 1024 x 512, where the y range of the chip is smaller than the beam size, and therefore will result in a slightly lower standard deviation than the x distribution.

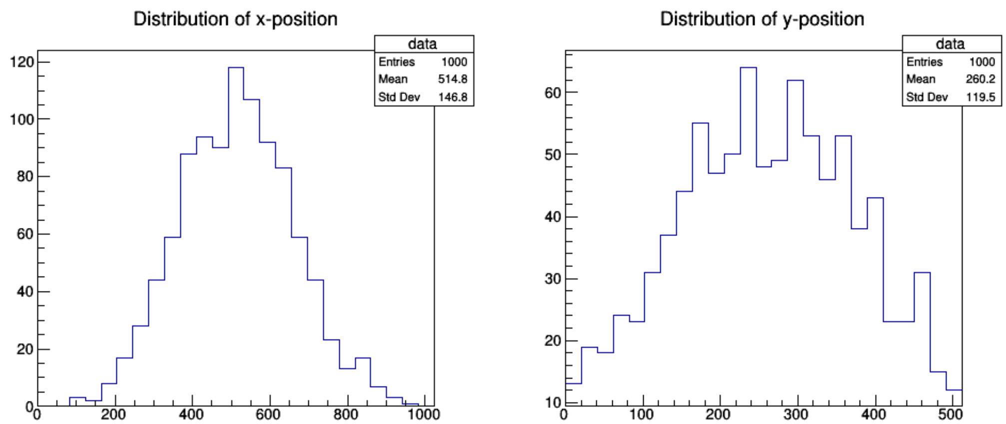
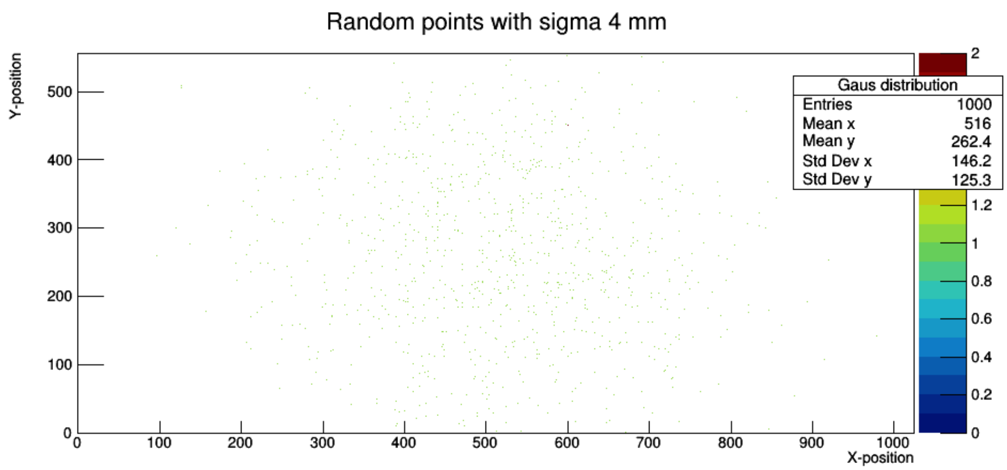


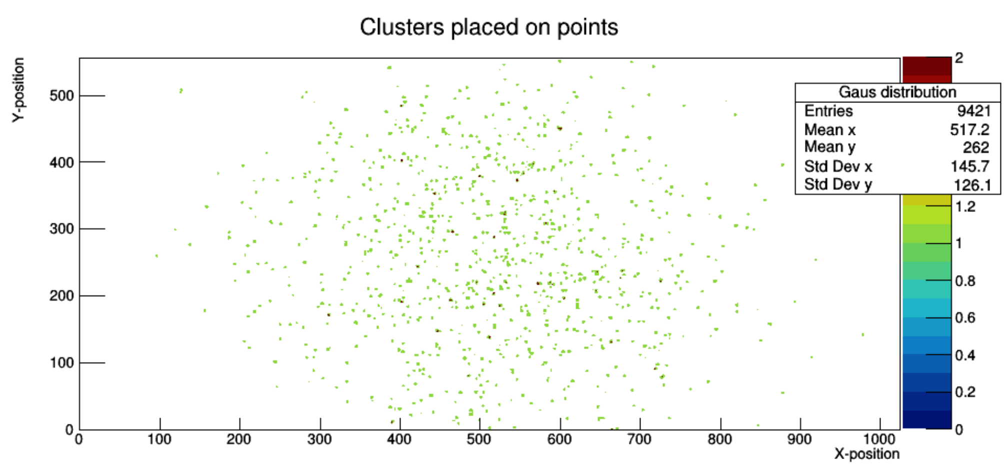

**Figure 3.12:** Distribution of randomly picked x and y pixel positions.

Figure 3.13a shows how the points are placed in the chip. Arbitrary clusters from the collection are placed onto random points as shown in Figure 3.13b. As a result, one will get a chip with 1000 clusters, and a chip occupancy of approximately 1.7%. In this example, the Gaussian beam size is set to 143 pixels in each lateral direction, which is approximately 4 mm.

Each pixel simulated in the detector will contain some information. This includes information about the position in the chip, and if one pixel is part of a multiple clusters. This information is not used in the method for finding a cluster, but in analysing the results of the cluster reconstruction.




**(a)**

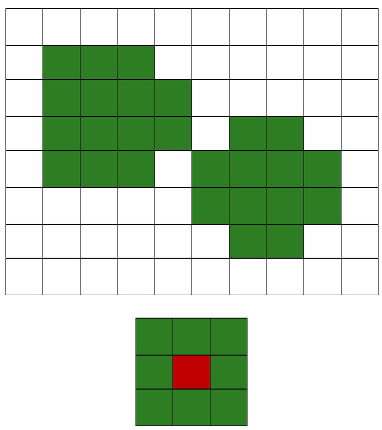



**(b)**

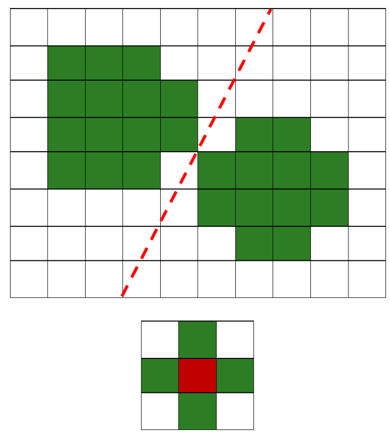
**Figure 3.13:** Simulation of helium particles hitting the ALPIDE chip. The beam size is a sigma of 4 mm in both x and y directions and 1000 particles are hitting the layer. There are used 1000 particles in the detector to clearly see the distribution of clusters.

### 3.2.2 Method for finding clusters

The method for finding clusters include going through the entire chip, pixel by pixel. When an active pixel is found, two approaches could be used. These are demonstrated in Figure 3.14. Figure 3.14a will find all activated pixels which are adjacent to one another in x, y, and diagonal directions. This means it will consider one pixel (the red one) and find all activated pixels around that pixel. It will keep doing this until there are no more adjacent activated pixels. Then, it classifies all the found pixels as one cluster. The approach shown in Figure 3.14b is similar, except it excludes the diagonal direction. This will result in the first approach classifying the simulated cluster as one cluster, and the other as two clusters in Figure 3.13. The method without diagonally adjacent clusters provides a better overall result, which is discussed in subsection 4.1.



**(a)** Find all activated pixels which are adjacent to one another in x, y and diagonal directions. Results in one cluster.



**(b)** Find all activated pixels which are adjacent to one another in x and y directions, not in a diagonal direction. Results in two clusters.

**Figure 3.14:** Algorithms which separate on two different terms when finding cluster in simulated chip.

The methods discussed above are simple techniques to reconstruct the clusters in the detector. The methods may be used to find the clusters in the detector when there is apparently no overlap between the clusters. However, when there are multiple overlaps among the clusters, the method will perform poorly. The source for misclassified clusters is the overlapping of two or more single clusters. In order to classify some of these overlapping clusters correctly, several cuts to recognise double clusters were developed. All the cuts use the above method without diagonal direction and will therefore be referred to as the basic method in this thesis.

### Cut 1 - Fill percentage

The first cut uses the fill percentage to distinguish whether a cluster identified by means of the basic method, should be divided into two clusters. The fill percentage means placing the cluster in a square or a rectangular box, and then calculating the percentage of active pixels that exist inside the box. In order to calculate the square box, one would take the largest height or weight and multiply it by itself. The rectangular box is calculated by multiplying height and width.

*Fill percentage = Activated pixels / box*

A separator value is thereafter chosen to distinguish when a cluster should be divided into two clusters. The fill percentage can be adjusted to a preferred value between 0 and 100%. When the cluster's fill percentage is less than the chosen value, the method will classify the identified cluster as two clusters instead of one. The double cluster will not be separated from each other, but they will be marked as two clusters.

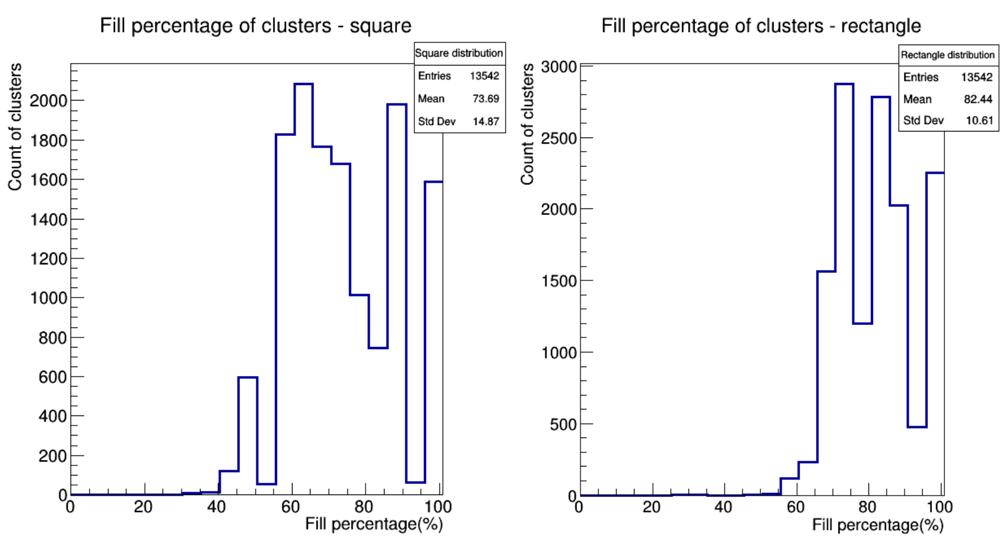



**Figure 3.15:** Distribution of fill percentage calculated with both square and rectangular box for clusters in data collection.

The distribution of the fill percentage in the data collection are presented in Figure 3.15. The square approach is shown in the left graph and the rectangular approach in the right. The graph on the left shows a broader spread in fill percentage and an average of 73.7%. The right graph, on the other hand, has a narrower distribution of the fill percentage and an average of 82.4%.

**Cut 2 - Asymmetry**

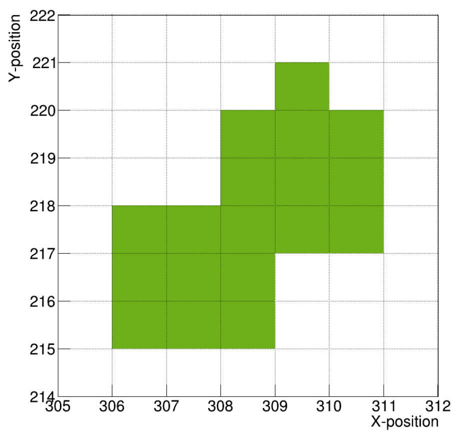
The second method is to make a cut based on the asymmetry of the cluster shape, where the distribution of pixels in the cluster is viewed in either the x or y direction. An example is shown in Figure 3.16. Figure 3.16a shows a double cluster in the detector created in simulation, which should be classified as two clusters. Figure 3.16b shows the distribution of pixels in x direction, and Figure 3.16c shows the distribution of pixels in a y direction. The method looks for deviations in the distribution in both directions, which results in asymmetric cluster shapes. The deviation can be:

1. Two clusters which are adjacent to one another where there exists a high pixel peak among lower ones in either the x or y distribution. The high pixel peak needs to be at least 2 pixels higher on both sides. An example of this is shown in Figure 3.16c. The peak contains 5 pixels, and the count of pixels on both sides of the peak contains 3 pixels. Since the difference in the count of pixels on both sides of a peak is 2, this fulfils the condition. The method will always compare 3 and 3 counts of pixels next to each other sequentially and look for a peak. If the peak is more than 1 pixel wide, it will not detect it.

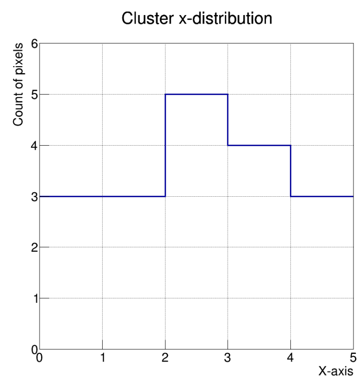
2. Opposite of the previous one, two clusters which are adjacent to one another where there exists a low pixel count among higher ones. Also here, 3 and 3 values of pixels are compared, if the first and the third value differ by at least 2 pixels to the second value, the condition is fulfilled.

3. Two clusters which are adjacent to one another with only one pixel in the middle of the distribution. For example, the distribution in y direction containing the following values: [3, 3, 2, 1, 3, 3, 2]. The middle value contains 1, which fulfils this condition.

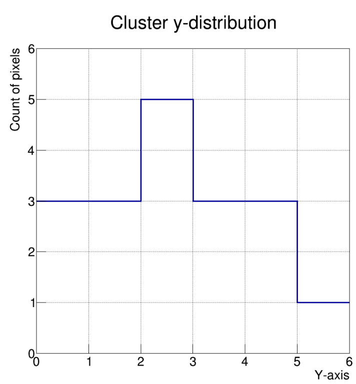
If at least one of these conditions for deviation in the distribution is fulfilled, the simulated cluster is marked as two clusters.




**(a)** A double cluster simulated in the detector.




**(b)** X distribution.




**(c)** Y distribution.

**Figure 3.16:** Distribution of cluster projected onto x and y axis.

### Cut 3 - Inactive pixels

Cut 3 uses almost the same method as used to remove clusters with holes as was described in subsection 3.1.2. It looks for continuously active pixels in both x and y direction. This method includes the outer edges of the cluster, in contrast to the method for removing holes in the cluster. An example of a double cluster which will be marked as two clusters is shown in Figure 3.17. When the method goes through column $x_1$ row by row, it will find an inactive pixel among active pixels at $y_5$. Therefore, the method will mark this cluster as a double cluster.

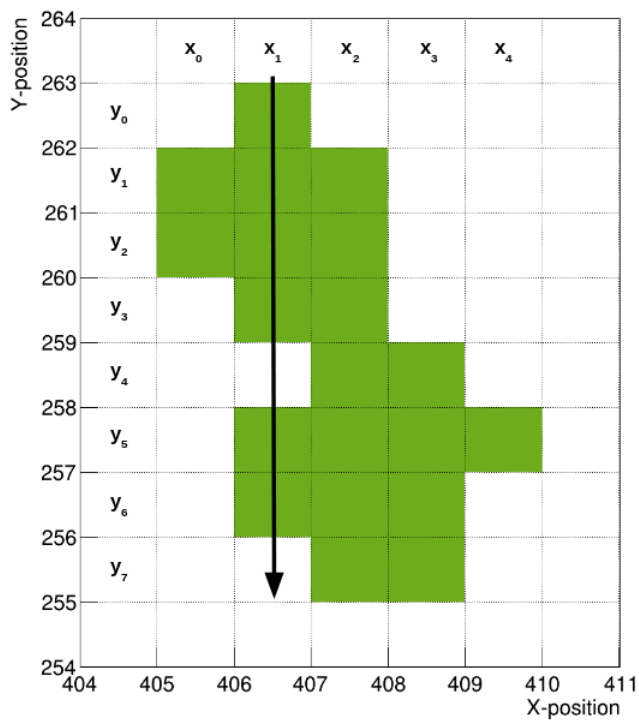



**Figure 3.17:** A double cluster which will be separated by the method used in approach 3 - inactive pixels.

#### Barrier (Separator) values

The three different cuts will also recognise a double cluster if the cluster exceeds the values for maximum size, height and width which were specified in subsection 3.1.2. The size is 30 pixels, and the height and width are 10 pixels. If the cluster exceeds these values, they will be classified as two different clusters.

Some clusters exist in the data collection which will be classified as 2 clusters by one of the cutting approaches, even though they are single clusters. To reduce this misclassification of clusters, a separator value is defined for the minimum size of a cluster. The cluster size will therefore need to exceed this value in order to use one of the cutting approaches on the cluster.

For example, if the fill percentage separator is set to 56%, then there are over 1100 single clusters which will be misclassified as two clusters in the data collection. However, if a minimum size separator value exists, of for example 9, the misclassified clusters will be significantly reduced. As a result, there will be around 150 clusters which will be misclassified in the data collection. Even though a few single clusters are misclassified, more double clusters will be able to be separated by the cut and the score of reconstructed clusters increased.

### 3.2.3 Method for analysing results

An analysis must be conducted to evaluate the results from the reconstructed clusters. The analysis determines which simulated clusters are perfectly classified and which are incorrectly classified.

The first analysis is performed on the method for reconstructing clusters with and without diagonal pixels. The method looks at each classified cluster. All the pixels in the cluster contain information from the simulation which will be used here. If one of the pixels in the cluster contains information about multiple clusters, or if the pixels belong to a different original cluster, then the entire cluster will be classified as an error. However, if all the pixels belong to the same cluster, and the size of the cluster is also the same as the original, it is classified as a perfectly reconstructed cluster.

The second analysis was performed on the extended method of reconstructed clusters with filter. If the method has classified a cluster as one cluster, it will go through the same procedure as above. However, if the method has classified the cluster as two clusters, it will look for only two original clusters in that exact cluster. If it is able to find only two clusters, it will be classified as two perfectly reconstructed clusters. On the other hand, if the cluster contains more than two clusters or only one cluster, it will be categorised as an error.

The analysis will in both cases classify the cluster as either a perfectly reconstructed cluster or a misclassified cluster. It will not take into account whether the cluster is nearly perfectly reconstructed, meaning that there are only one or a few pixels more than the origin cluster.

# Chapter 4

# Results and evaluation

This chapter will answer the research question about which method will perform best in terms of efficiency compared to particles per pencil beam. The efficiency is defined to be the radio of perfectly reconstructed clusters to total number of cluster in the detector. The basic method identified clusters which were adjacent in x and y directions, and then different cuts were tested in order to distinguish single clusters from overlapping clusters in the detector.

The result of each cut was thereafter evaluated by looking at how many perfectly reconstructed clusters were found and how many clusters there were in the detector. The desired perfectly reconstructed clusters percentage is as high as possible, and at least 96%. The percentage needs to be high because there are a lot of hits in the calorimeter in the planned proton CT system in Bergen. The calorimeter contains approximately 40 layers, and if around 4% are lost in every detector layer, then the effective per-track efficiency would be $96\%^{40} \simeq 20\%$. This is a high total loss, which would results in a significantly reduced precision and increase the radiation dose to the patient. However, methods have been developed to mitigate the loss in the track reconstruction part of Proton CT. If 100 particles hit the calorimeter, it is expected that there will be 88% correctly reconstructed proton paths without taking the effective per-track efficiency into account, and 85% when considering them[6]. This is before applying the filters discussed in chapter 3. Therefore, we will fortunately get much better than 20% per-track clustering efficiency.

The effect of having as many particles per pencil beam as possible, with a acceptable efficiency, is making the duration for taking a proton CT shorter for the patient.

In addition, in order to to evaluate the performance of the different cuts in terms of particles per pencil beam and efficiency, the misclassified clusters from each cut were analysed. The misclassified clusters are, as described before, discarded clusters which did not fulfil the criteria for perfectly reconstructed clusters (subsection 3.2.3). The misclassified clusters were divided into 3 categories:

- Category 1 - Found 2, was 3+: The cut classified the cluster found using the basic method as 2 clusters, but there were at least 3 clusters originally.
- Category 2 - Found 2, was 1: The cut classified the cluster found using the basic method as 2 clusters, but there was only a single cluster originally.
- Category 3 - Found 1, was 2+: The cut found only 1 cluster, but there were at least 2 clusters originally.

All the different cuts will only distinguish between a single cluster and 2 clusters. Therefore 3 or more overlapping clusters will never be classified with these cuts. These discarded clusters will fall into category 1. Category 2 contains single clusters which were misclassified as 2 double clusters because it met the criteria for abnormal clusters for that cut. The last category, category 3, is the division into which most misclassified clusters fall. The cut was not able to classify each cluster as an abnormal cluster and therefore only found a single cluster.

The misclassified clusters from each cut are also analysed where one looks at how many extra pixels each cluster contains. This information tells us how many clusters were nearly perfectly reconstructed.

The following subsection will present the step-by-step results of each cut described in subsection 3.2.2. Firstly, how the different parameter values were chosen. Followed by the results of each cut. Thereafter which type of misclassified clusters the cut had. Finally, all the cuts are combined in different variants to compare which combinations perform best.

Each of the simulations contains the following settings if nothing else is specified. The beam size correspond to a circular Gaussian with a standard deviation of 4 mm, or approximately 143 pixels. The simulation is conducted with 0-500 clusters placed in a detector layer, with a step of 25. Each step shows an average of 100 simulations in order to get better statistical significance. The graphs of misclassified clusters include these settings as well.

## 4.1 Results from the basic method

The results from the first two methods described in subsection 3.2.2 are presented in Figure 4.1. The methods identified clusters either by finding pixels in the detector which were adjacent in x, y and diagonal directions or only x and y directions (basic method). The graph shows how many clusters the two methods were able to reconstruct with an increase in the number of clusters in the detector. The y axis shows the percentage of perfectly reconstructed clusters. The x axis shows how many clusters there were in the detector in each simulation.

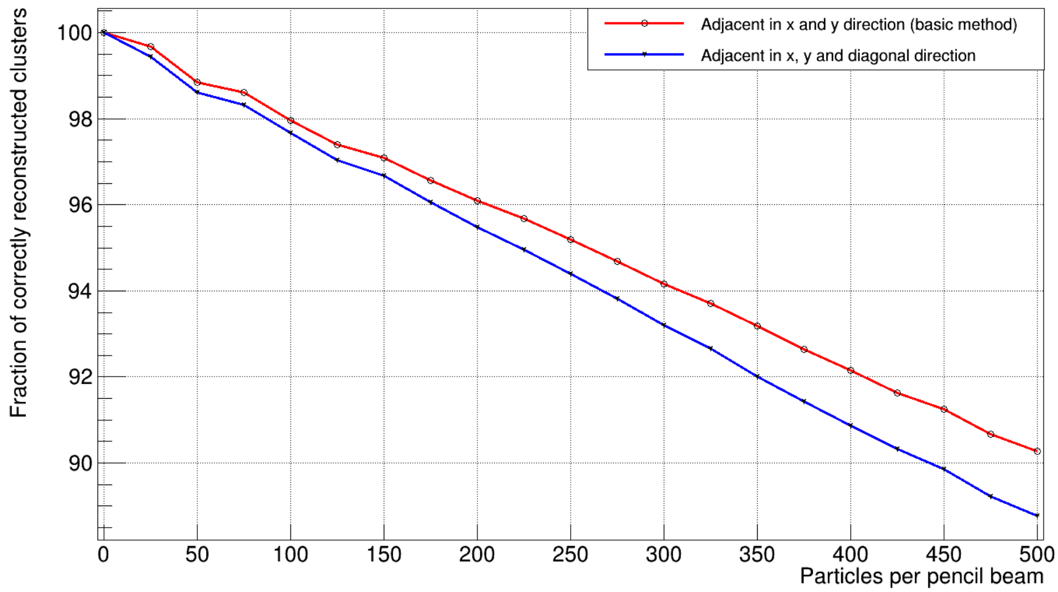



**Figure 4.1:** Results finding clusters with and without diagonal directions using a beam size of 4 mm.

Figure 4.1 shows that the basic method resulted in the best score of perfectly reconstructed clusters. The difference in percentages between the two methods increases slightly with the clusters' occupancy in the detector. Figure 4.1 shows that in order to have a minimum of 96% given a beam size of 4 mm, there cannot be more than approximately 175-200 clusters in the detector depending on the method.

### 4.1.1 Misclassified clusters

Figure 4.2 illustrates the average errors of all the misclassified clusters using the basic method with increasing clusters in the detector. The method has no cuts to distinguish a single cluster from multiple clusters, and therefore it will only have errors of category 3.

The misclassified clusters were also analysed in terms of how many extra pixels each cluster had, which is shown in Figure 4.3. The graph shows around

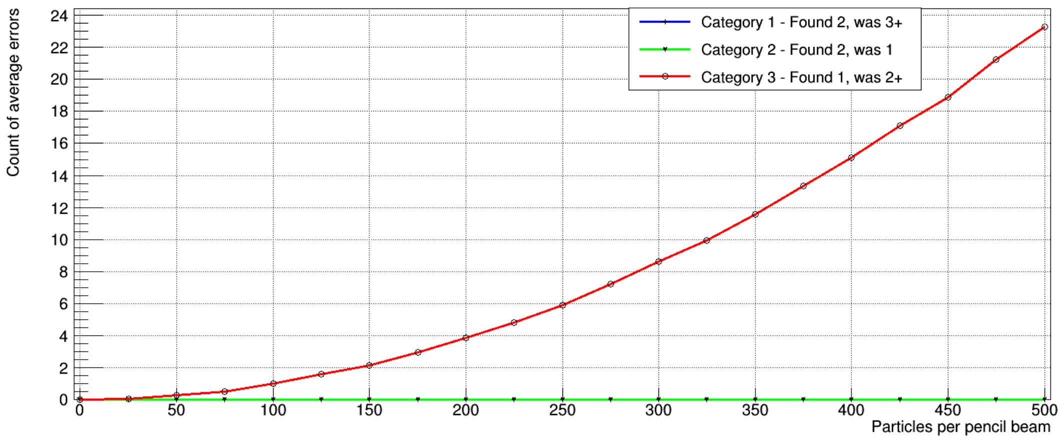



**Figure 4.2:** The average errors of misclassified clusters in the simulation.

35 000 misclassified clusters out of 525 000 clusters in the entire simulation, which is approximately 6.67% errors. The x axis shows how far from perfectly reconstructed each cluster was. One can see that there are approximately 2000 misclassified clusters which fall into 0-2 extra pixels. These errors can be considered as nearly perfectly reconstructed clusters. Over 500 clusters with 0 extra pixels have either a fully overlapping cluster, or it is a larger cluster which covers the overlapping cluster completely. The shape of the graph (figure 4.3) is quite similar to the size distribution of size in the cluster collection (Figure 3.16a), and both of the graphs have an average of around 9 pixels.

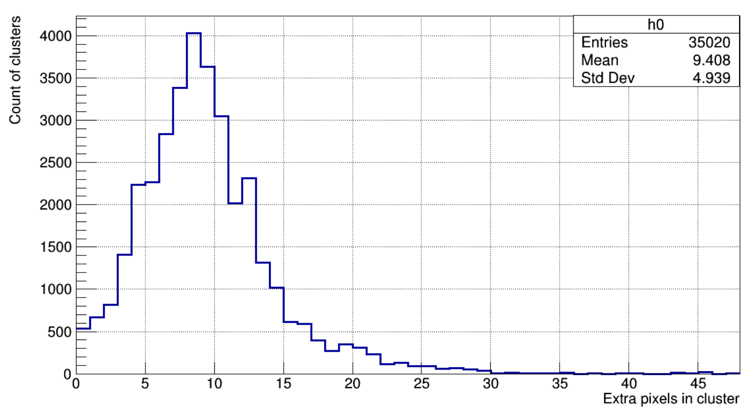



**Figure 4.3:** How many extra pixels each misclassified clusters had, using the basic method.

### 4.1.2 Results from different beam sizes

The size of the beam determines how centred or spread the clusters are placed in the detector. A smaller beam size will centre the clusters and result in more overlap between clusters than a larger beam size. The beam size is normally set to a range between 71 pixels and 214 pixels in x and y directions, which is the same as 2 mm and 6 mm. There is not much difference in cluster overlapping when the beam size exceeds approximately 6 mm (214 pixels). The result of different beam sizes is shown in Figure 4.4 where one can see a cluster range from 0-500 with a step of 25. The beam sizes presented in the graph are 2-6 mm and the basic method was used.

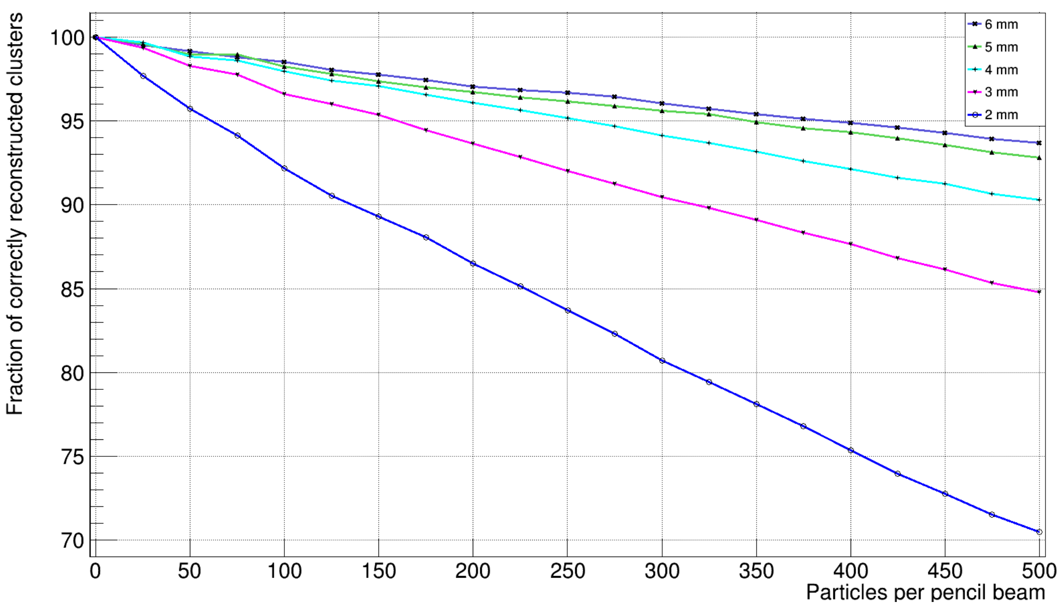



**Figure 4.4:** Results from simulation with different sigma values using the basic method - where pixels are adjacent in x and y directions.

One can see that the result of perfectly reconstructed clusters varies depending on the beam size. A beam size of 2 mm, which is the dark blue line on the graph, decreases rapidly compared to the others. This is due to a lot of overlapping of clusters in the detector, which may be difficult to separate from one another. Additionally, the graph shows that a bigger beam size results in a higher fraction of perfectly reconstructed clusters which is due to a higher spread of clusters in the detector. This simulation was conducted to examine the effect of different beam sizes and the performance was found to vary.

## 4.2 Results from cut 1 - fill percentage

The fill percentage cut which is described in subsection 3.2.2, uses the fill percentage of a cluster placed in a square or rectangular box to determine if the cluster should be defined as two clusters. Additionally, one has the minimum cluster size barrier value, which decides the size a found cluster must be in order to be classified as two clusters by the fill percentage cut.

Both the fill percentage approaches have their parameters selected according to the following conditions: Simulations with 100, 150, 200, 250 and 300 clusters in the detector were conducted with various fill percentages and minimum sizes. The simulations are carried out using a beam size of 143 pixels (4mm) and minimum size of 0, 7-17. The fill percentage varied depending on the approach. The square box approach had a fill percentage between 40-70%, and the rectangular approach had between 55-80%. Each value in the tables is an average result of 100 simulations with different clusters placed in the detector.

The results from the square box approach are shown in details in Appendix A including Table 1, 2, 3, 4 and 5 with 100-300 clusters in the detector. The results are summarised in Table 4.1. Figure 4.5 demonstrate the results when changing the parameters with 200 clusters in the detector. The figure shows that if the fill percentages are increased, the minimum size value should also be increased.

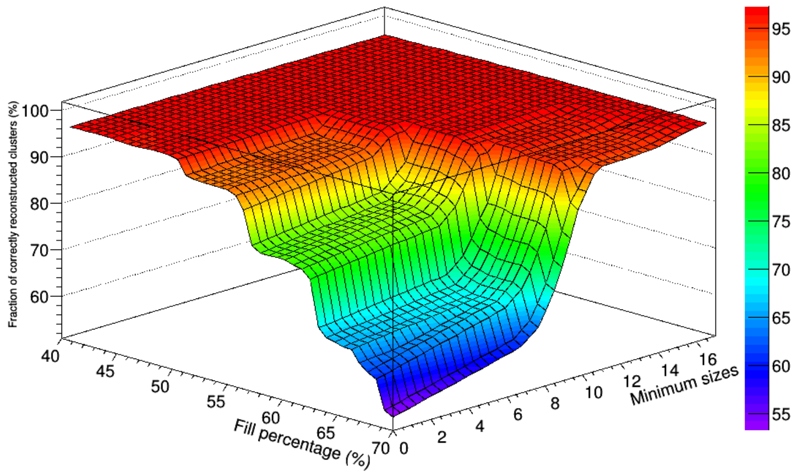



**Figure 4.5:** Results tuning the parameters for the fill percentage square approach using 200 clusters in the detector.

| Summarised results | | | | |
|---|---|---|---|---|
| Cluster in detector | Fill percentage | Minimum size | Highest score | Difference from next highest score |
| 100 | 49<br>50 | 13<br>13 | 98.520% | 0.010% |
| 150 | 51<br>52<br>53<br>54<br>55<br>51<br>52<br>53<br>54<br>55 | 13<br>13<br>13<br>13<br>13<br>15<br>15<br>15<br>15<br>15 | 97.800% | 0.007% |
| 200 | 54<br>55 | 13<br>13 | 97.265% | 0.015% |
| 250 | 54<br>55 | 13<br>13 | 96.896% | 0.016% |
| 300 | 56 | 13 | 96.457% | 0.014% |

**Table 4.1:** Summarised results for combining different minimum sizes and fill percentage (square box) with different amount of clusters in the detector. There are two combinations which performs best aggregated. The combinations are fill percentage 54% and 55% combined with a minimum size of 13.

Table 4.1 shows that all the best scores had a minimum value of either 13 or 15, and a fill percentage between 49%-56%. Recurrent combinations were fill percentages of 54% and 55% combined with a minimum size of 13. To determine which combination to use, a further simulation was conducted. It consisted of simulations with a beam size of 143, 0-500 clusters in the detector, with a step of 25 clusters, and each step shows an average of 100 simulations. The result of the different fill percentages was identical. New simulations were also conducted with an average of 200 simulations instead of 100, which resulted in the same results. Since both of the fill percentage values performed identically, one of them was simply chosen. The parameters for the fill percentage square approach were therefore chosen to be a fill percentage of 55% and a minimum size of 13.

The results from each simulation run for selecting the fill percentage rectangular approach are presented in Appendix B. This includes details from Table 6 (100 clusters), 7 (150 clusters), 8 (200 clusters), 9 (250 clusters), 10 (300 clusters), and are summarised in Table 4.2. An illustration of the results from 200 clusters in the detector is also presented in Figure 4.6.

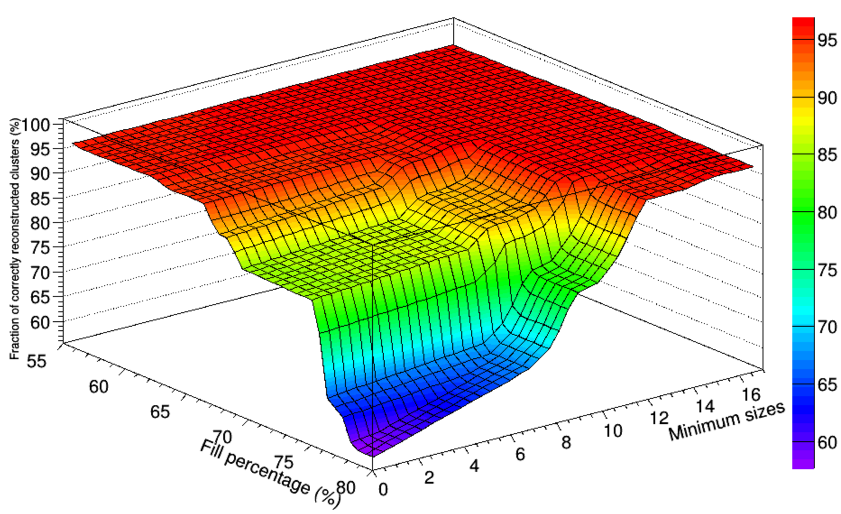



**Figure 4.6:** Results from combining different variants of minimum size and fill percentage calculated with rectangular box approach using 200 clusters in detector.

| Summarised results | | | | |
|---|---|---|---|---|
| Cluster in detector | Fill percentage | Minimum size | Highest score | Difference from next highest score |
| 100 | 68 | 15 | 98.300% | 0.020% |
| 150 | 72 | 14 | 97.580% | 0.070% |
| 200 | 72<br>68 | 14<br>13 | 96.930% | 0.015% |
| 250 | 72 | 14 | 96.572% | 0.024% |
| 300 | 72 | 14 | 95.967% | 0.037% |

**Table 4.2:** Summarised results for combining different minimum sizes and fill percentage (rectangle box) with different amount of clusters in the detector. The best combination is fill percentage 72% combined with a minimum size of 14.

Table 4.2 shows all the best scores from various counts of clusters in the detector. The best scores had a fill percentage of either 72% or 68% and minimum sizes of 13, 15 or 17. The most recurrent combination was a fill percentage of 72% and minimum size of 14, and was therefore selected as the best parameter for this cut.

The two best combinations between fill percentage and the minimum size are compared with the basic method in Figure 4.7. The basic method is better using between 0-50 clusters in the detector, and the fill percentage cuts perform better with 50 and more clusters in the detector. The graph also shows that the square approach performs slightly better than the rectangular approach. The reason why the square approach is better than the rectangular approach is that most clusters have a circular shape. Therefore, the square approach will more easily detect when two clusters are overlapping. The cut with the square approach can have nearly 350 clusters in the detector in order to obtain a result higher than 96% of perfectly reconstructed clusters. The rectangular approach, on the other hand, can only have approximately 300 clusters in the detector.

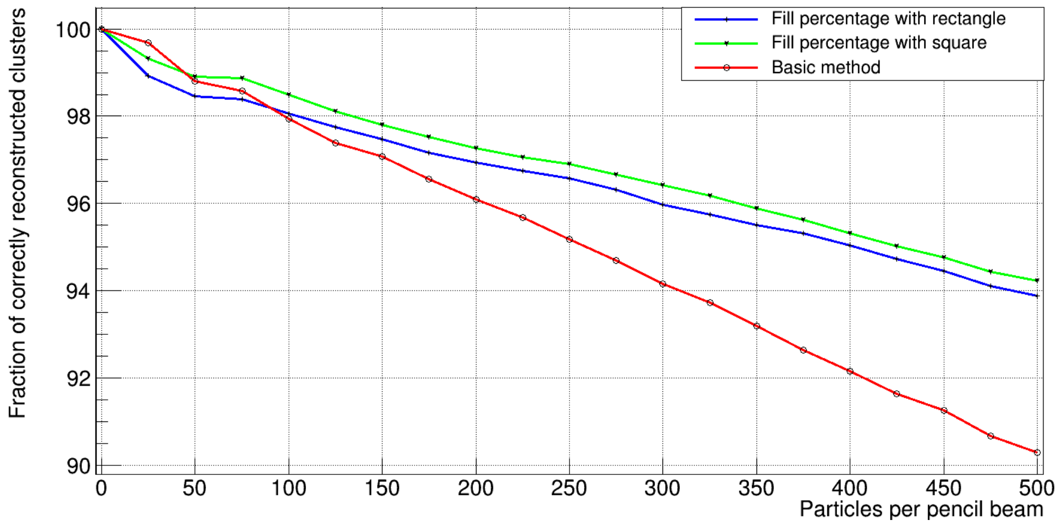
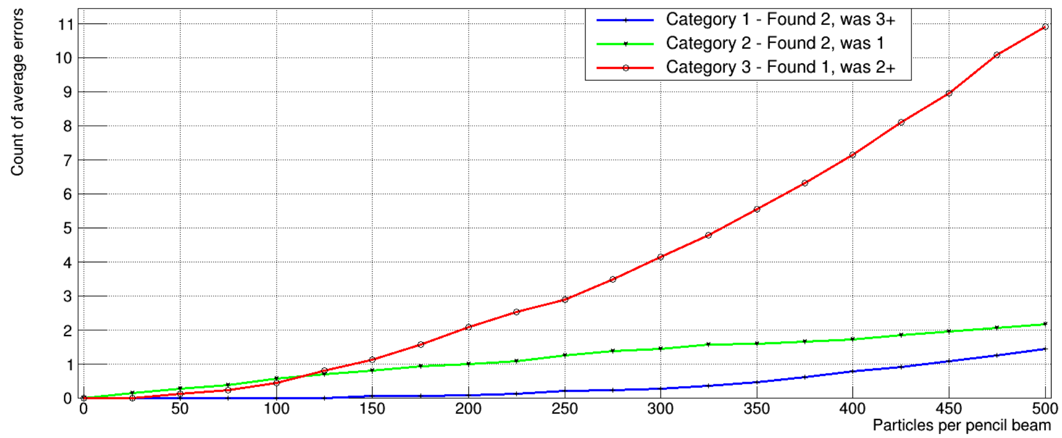



**Figure 4.7:** Results from simulations with fill percentage calculated using square and rectangular box approaches compared with the basic method. The fill percentage calculated with a square box approach is 55% and minimum size of 13, and the fill percentage calculated with a rectangular box is 72% and minimum size of 14.

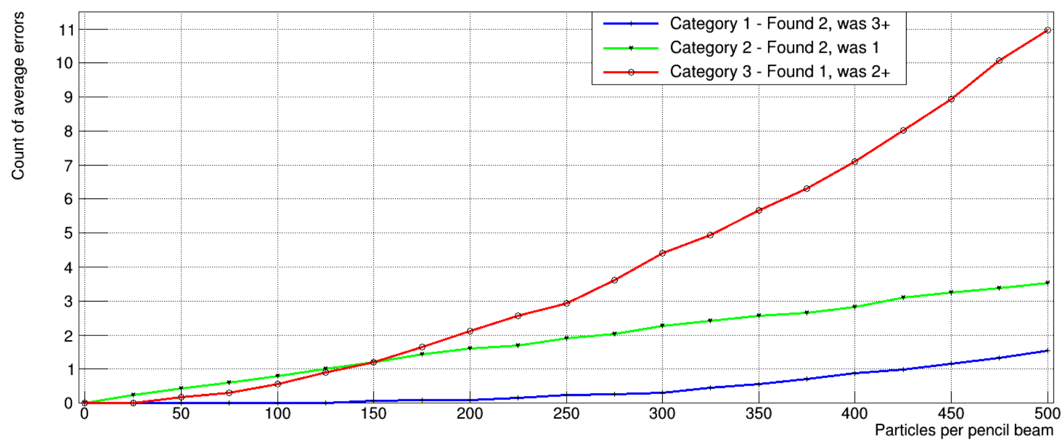
### 4.2.1 Misclassified clusters

Figure 4.8 shows which categories the different fill percentage cuts divide the misclassified clusters into. Category 1 and category 3 are not far from identical in Figure 4.8a and 4.8b. Category 2 on the other hand, varies a lot more. The rectangular cut results in approximately twice as many average errors for category 2 than the square cut. This is why the square cut performs slightly better at finding perfectly reconstructed clusters.

An illustration of how many extra pixels each misclassified cluster had using the two cuts is shown in Figure 4.9. In contrast to the basic method, both of the fill percentage cuts have a high peak at 0 extra pixels in the cluster. This is the result of misclassifying single clusters, which were shown in Figure 4.8, category 2. In other words, the peak includes mostly clusters which were classified by the cut as two clusters but were only one cluster.



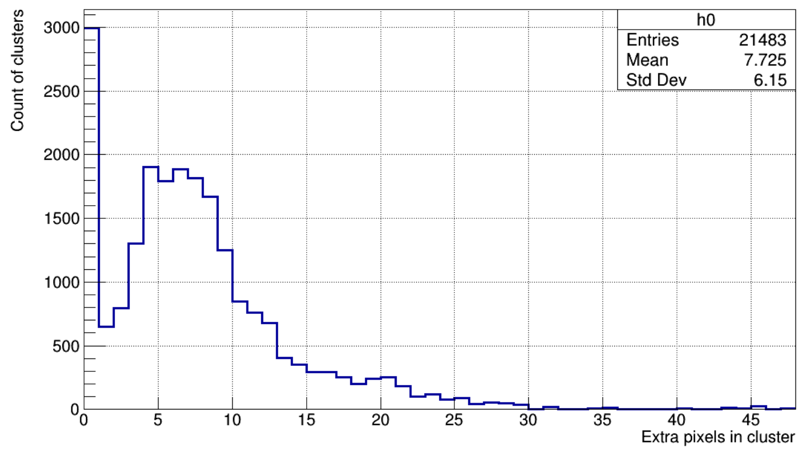

**(a)** Fill percentage square cut.



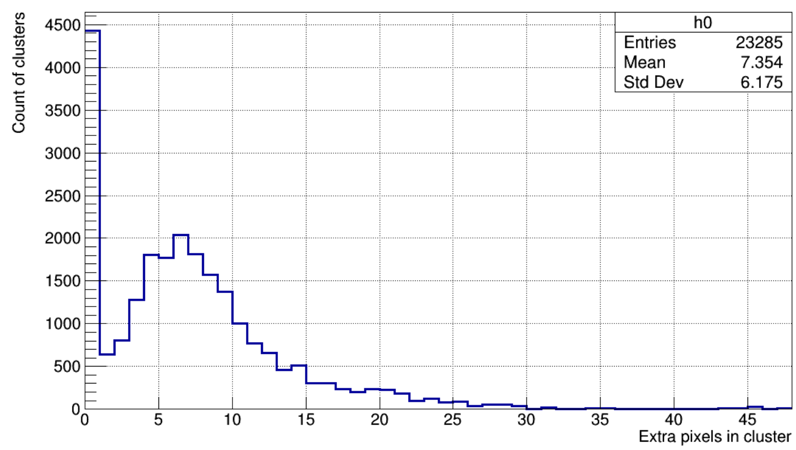

**(b)** Fill percentage rectangular cut.

**Figure 4.8:** Types of misclassified clusters for the fill percentage cuts.

As described earlier, the basic method had around 35 000 misclassified clusters out of a total of 525 000. Both of the fill percentage cuts have significantly lower total misclassified clusters. The cut with the square approach beats the rectangular approach with approximately 1800 fewer misclassified clusters.




**(a)** Fill percentage square cut.




**(b)** Fill percentage rectangular cut.

**Figure 4.9:** Distribution of how many extra pixels each misclassified cluster had, using different fill percentage cuts.

## 4.3 Results from cut 2 - Asymmetric

The asymmetric cut which is discussed in subsection 3.2.2, looks at each identified cluster and views the distribution of pixels in x and y direction to determine if the cluster should be classified as one or two clusters. The results from this cut are presented in Figure 4.10, along with the basic method and the asymmetric cut combined with a minimum size value.

The minimum size value was determined by combining cut 2 and different minimum sizes. It was conducted with a minimum size values of 0 and 7-17. The results of how many perfectly reconstructed clusters the different combinations had are presented in Table 11 in Appendix C. The minimum size values 11 to 15 gave very similar results. In order to determine the best combination of these minimum size values, the average for each minimum value is calculated, and the value which gave the highest percentage result was chosen. The results are presented in Table 4.3. The value which gave the highest percentage of perfectly reconstructed clusters were 13.

| Minimum size value | Average result (%) |
|---|---|
| 0 | 92.076 |
| 7 | 92.563 |
| 8 | 95.970 |
| 9 | 96.059 |
| 10 | 96.301 |
| 11 | 96.401 |
| 12 | 96.451 |
| 13 | 96.501 |
| 14 | 96.481 |
| 15 | 96.445 |
| 16 | 96.272 |
| 17 | 96.146 |

**Table 4.3:** Average results from simulation of asymmetric cut with different minimum size values (Appendix C, Table 11). The minimum size of 13 gave the highest score.

Figure 4.10 shows that the asymmetric cut performs poorly without the minimum size value, even worse than the basic method. The basic method is best with 0-50 clusters in the detector, and thereafter the asymmetric cut combined with a minimum size value of 13 performs better. The best method (blue line) can have up to around 275 clusters in the detector in order to have a higher percentage than 96% of perfectly reconstructed clusters.

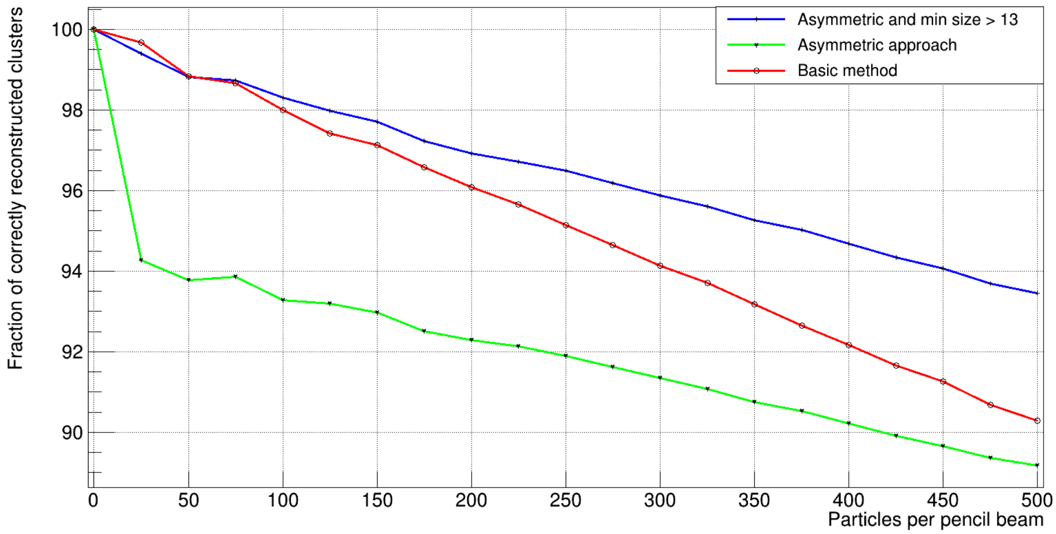



**Figure 4.10:** Results from asymmetric cut compared with the basic method.

### 4.3.1 Misclassified clusters

The misclassified clusters are presented in different categories in Figure 4.11. The cut results in a few errors of category 1 and 2, and a higher amount of category 3 errors. If one compares category 3 in this figure with the errors for the fill percentage cuts, one can see that this cut has a higher number of misclassified clusters than both of the fill percentage cuts. However, this cut has fewer misclassified clusters of category 2 than the fill percentage cuts.

Figure 4.12 shows the distribution of extra pixels on each misclassified cluster which was due to overlapping or wrongly classifying single clusters. There were approximately 24 400 misclassified clusters out of 525 000 simulated clusters. This is higher than both of the fill percentage cuts, but still significantly better than the basic method. The asymmetric cut also has a high peak at 0 extra pixels, as both of the fill percentage cut had. However, the peak is much lower than the fill percentage peaks and are on the same level as the second peak on the graph. This is due to fewer misclassification of single clusters in this cut in comparison to the fill percentage cuts.

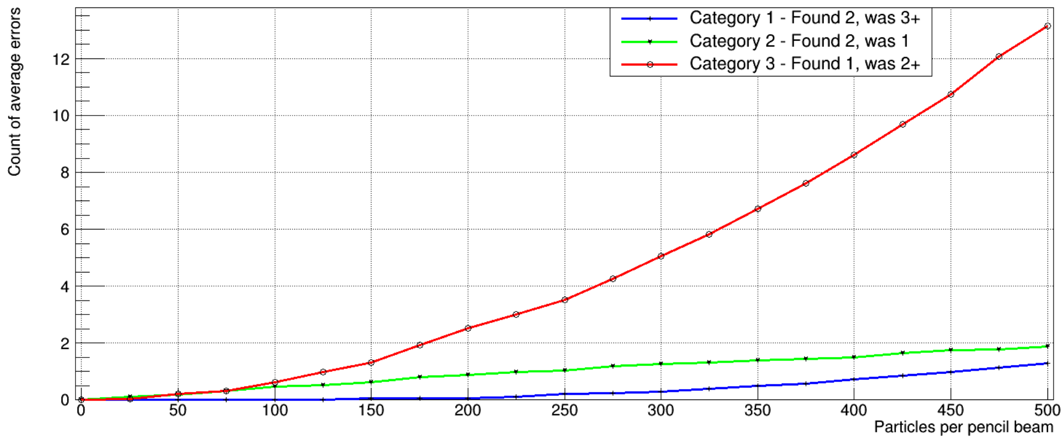



**Figure 4.11:** Average errors of misclassified in simulation for the asymmetric cut.

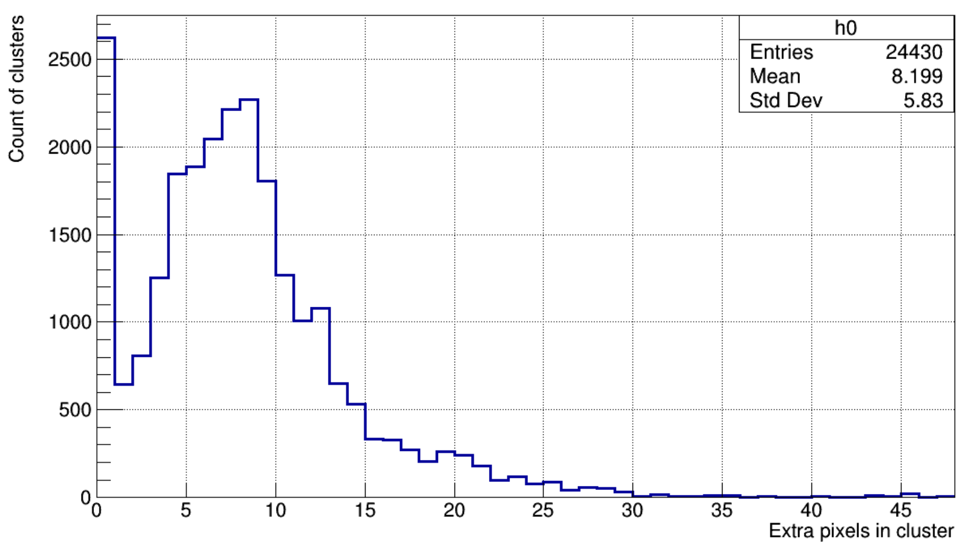



**Figure 4.12:** Distribution of how many extra pixels each misclassified cluster had, using the asymmetric cut.

## 4.4 Results from cut 3 - Inactive pixels

The inactive pixels cut, which is described in subsection 3.2.2, looks for continuously active pixels in both x and y directions in each cluster. When it finds inactive pixels among the active ones, the cluster is classified as two clusters. This cut was also tested together with the minimum size value, similar to the asymmetric cut. The simulation was the same as in subsection 4.3, except that the inactive pixels cut were used instead. However, the minimum size value seemed to have little effect on this cut, as can be seen in Table 4.4. The table is calculated in the same way as Table 4.3, and the full table of the simulation is available in Appendix D - Table 12. Since the minimum size value had a very small effect on the method, the value 0 was chosen.

| Minimum size value | Average result(%) |
|---|---|
| 0 | 96.261 |
| 7 | 96.264 |
| 8 | 96.264 |
| 9 | 96.264 |
| 10 | 96.261 |
| 11 | 96.253 |
| 12 | 96.118 |
| 13 | 96.252 |
| 14 | 96.215 |
| 15 | 96.180 |
| 16 | 96.106 |
| 17 | 95.984 |

**Table 4.4:** The average result from simulation of inactive pixels cut with different minimum size values (Appendix D, Table 12). The minimum size values resulted in nearly the same results.

Figure 4.13 shows the result of inactive pixels cut compared with the basic method. One can see that the inactive pixels cut performs better overall than the basic method. One can have up to 275 clusters in the detector to obtain a higher result than 96% of perfectly reconstructed clusters.

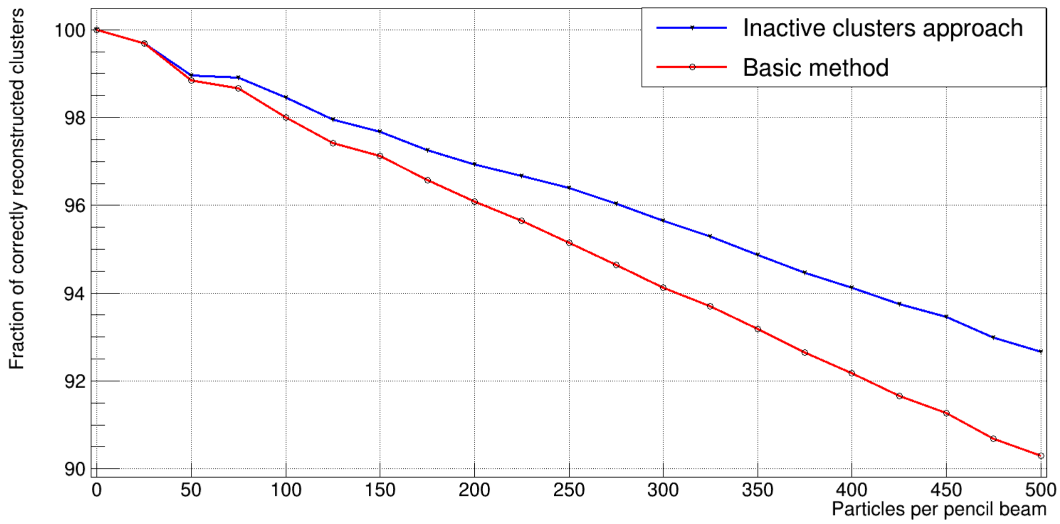



**Figure 4.13:** Results from inactive pixels cut compared with the basic method.

### 4.4.1 Misclassified clusters

The misclassified clusters divided into 3 categories are shown in Figure 4.14. The majority of errors falls into category 3, which is clusters that were classified as single clusters, but were at least 2 clusters. This cut has overall higher average errors in this category than the other cuts. However, both of the other categories for this cut have lower average errors than all the other cuts.

Each misclassified cluster is also compared with how many extra pixels it was found with. The distribution is presented in Figure 4.15. The shape of the graph is quite similar to the distribution of the basic method, except that it has fewer errors in total. The cut gave approximately 26 600 misclassified clusters out of 525 000 clusters. This is a higher value than the other cuts, but it is still considerably better than the basic method.

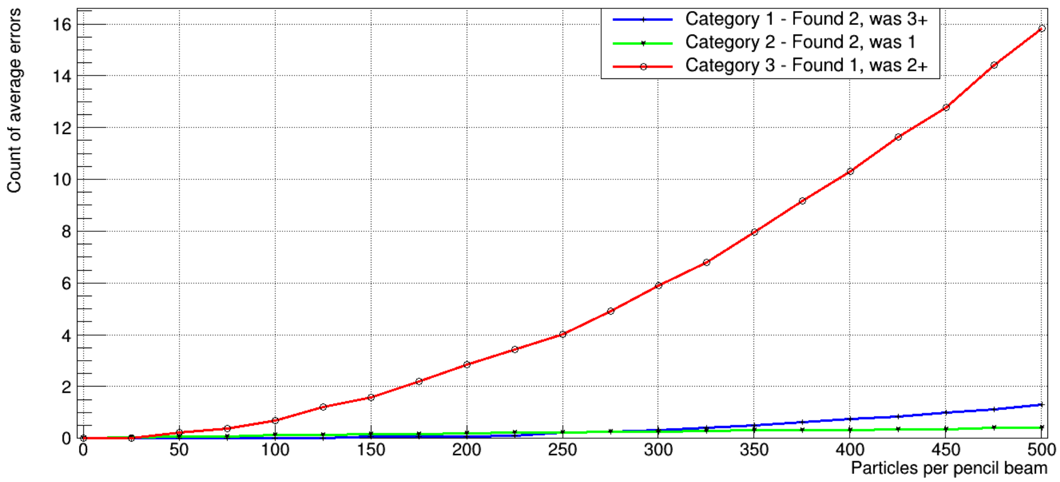



**Figure 4.14:** Average errors of misclassified clusters in simulation.

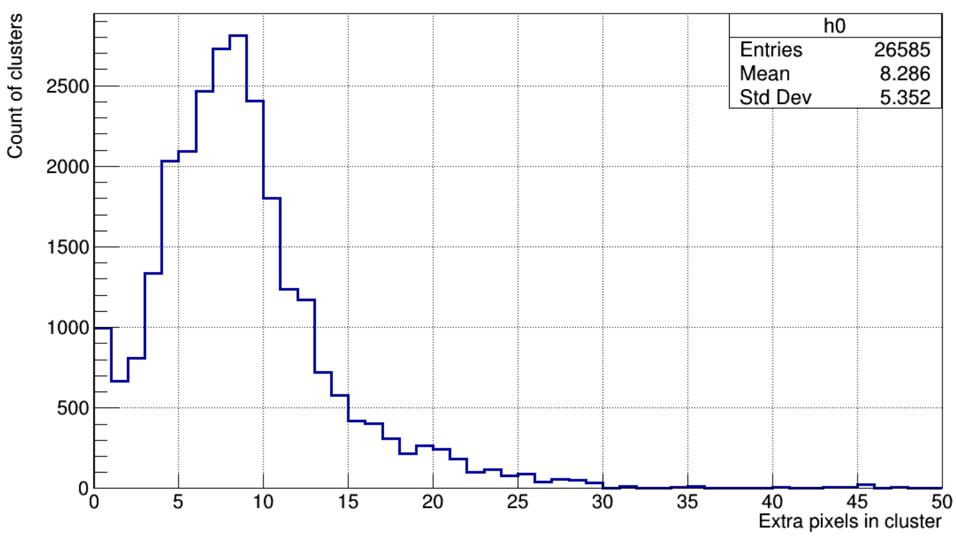



**Figure 4.15:** Distribution of how many extra pixels each misclassified cluster had, using the inactive pixels cut.

## 4.5 Combinations of different cuts

All the cuts were also combined in different combinations to see if any of the combinations would perform better than each cut separately. The results of these combinations are presented in Figure 4.16, and contains an average result of 200 instead of 100 simulations. This is to determine the optimal cuts with an even higher statistical significance.

The parameters for each combination are not fully optimised, but rather chosen based on the best parameters for each separate cut. The fill percentage value varies depending on the minimum size value, and therefore these settings are chosen over the settings of other cuts when combined with the fill percentage cut. In addition, the minimum value for the asymmetric cut is chosen when combining this cut and the inactive pixels cut. This is because the inactive pixels did not vary much in the results when using different values, but the asymmetric cut, on the other hand, improved significantly when setting the minimum size value.

Figure 4.16 shows the results of 8 different ways to cut clusters. After passing 150 clusters in the detector the different combinations roughly fall into 3 groups. The superior group includes different combinations of cuts combined with the fill percentage cut. In order to have a perfectly reconstructed clusters percentage of above 96%, one can have approximately up to 350 clusters in the detector by choosing one of them. The intermediate group includes the inactive pixels cut, asymmetric cut and a combination of these two. These, on the other hand, can have up to 250-275 clusters in the detector to have a score of at least 96%. The last group, which performed poorest, was the basic method and cannot have more than 200 clusters in the detector to reach above 96%. The different cuts will result in a variation of 200-350 clusters in the detector to obtain at least 96% perfectly reconstructed clusters. Even 200 clusters in the detector will give a higher efficiency than the track reconstruction conducted in [6] by the Bergen proton CT research team.

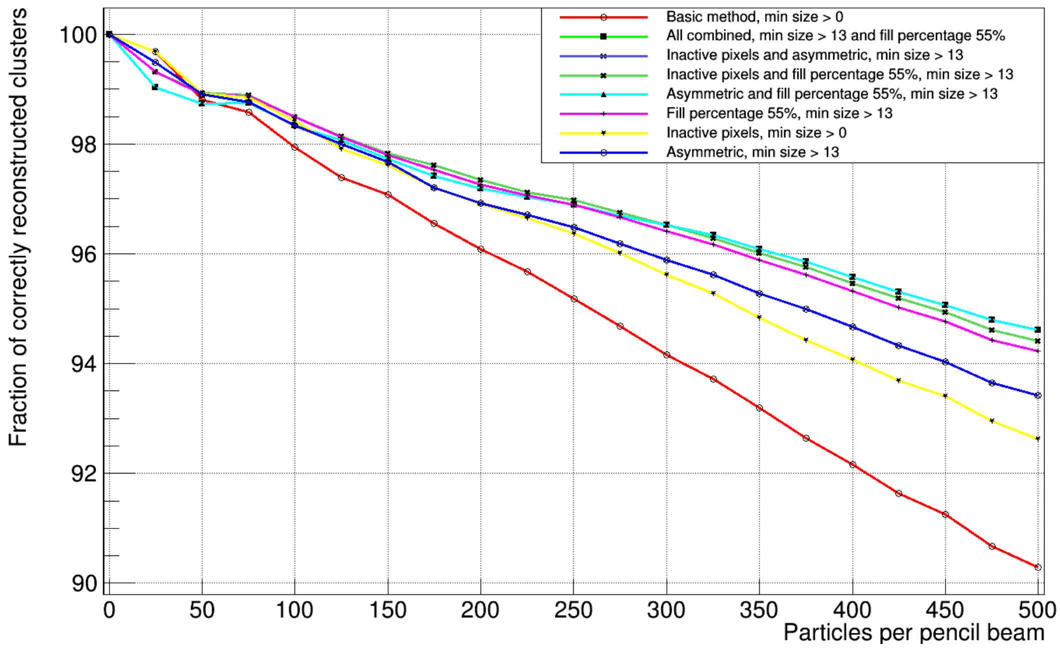



**Figure 4.16:** Results from simulation using different cuts with a beam size of 143 pixels and an average result of 200 simulations per point.

### 4.5.1 Misclassified clusters

A lot of the combinations resulted in similar graphs for misclassified clusters. All cuts combined with the fill percentage cut resulted in nearly identical graphs as the fill percentage graphs (Figures 4.8 and 4.9) for misclassified clusters. In addition, the combination of the asymmetric and inactive pixels cut has almost identical graphs for misclassified clusters as the asymmetric cut (Figures 4.11 and 4.12).

## 4.6 The optimal cuts

The inactive pixels cut gave the best score of all the combinations of cuts presented in this thesis from 0-100 clusters in the detector, which was shown in Figure 4.16. Subsequently, all the combinations of the fill percentage cut performed quite similarly with 100-500 clusters in the detector, but there were some minor differences. Figure 4.16 together with Table 13 presented in Appendix E, shows that the optimal cut for 100-300 clusters is the combination of the inactive pixels cut and the fill percentage cut. The optimal cuts for 300-500 clusters are the combination of the asymmetric cut and the fill percentage cut, and all cuts, which performs identically. There is no effect when including the inactive pixels cut, with the fill percentage and asymmetric cuts. Therefore, the optimal cut for 300-500 clusters is the combination of the fill percentage and the asymmetric cuts. The optimal cuts for different clusters in the detector are summarised in Table 4.5.

All the cuts combined with the fill percentage cut were not fully optimised by parameter tweaking for each combination. The differences are very small, and the optimal result could be different if the parameters were tuned for each combination.

| Combination of optimal cuts | Clusters in detector |
|---|---|
| Inactive pixels | 0-100 |
| Fill percentage and inactive pixels | 100-300 |
| Fill percentage and asymmetric | 300-500 |

**Table 4.5:** Results of optimal cuts depending on the number of clusters in detector.

### 4.6.1 How many clusters the optimal cuts identified

Figure 4.17, 4.18 and 4.19 shows how many clusters the optimal cuts identified in the detector (blue line). Additionally, the graph includes how many perfectly reconstructed clusters (red line) it found, with error bars of 1 standard deviation.

Both of the combined cuts (Figure 4.18 and 4.19) classifies several original single clusters as two clusters (category 2). As a result, the cuts found more than 100% of clusters in the detector. The inactive pixels cut, on the other hand, classified fewer of these errors. This is the reason why the inactive pixels cut performs better with few clusters in the detector.

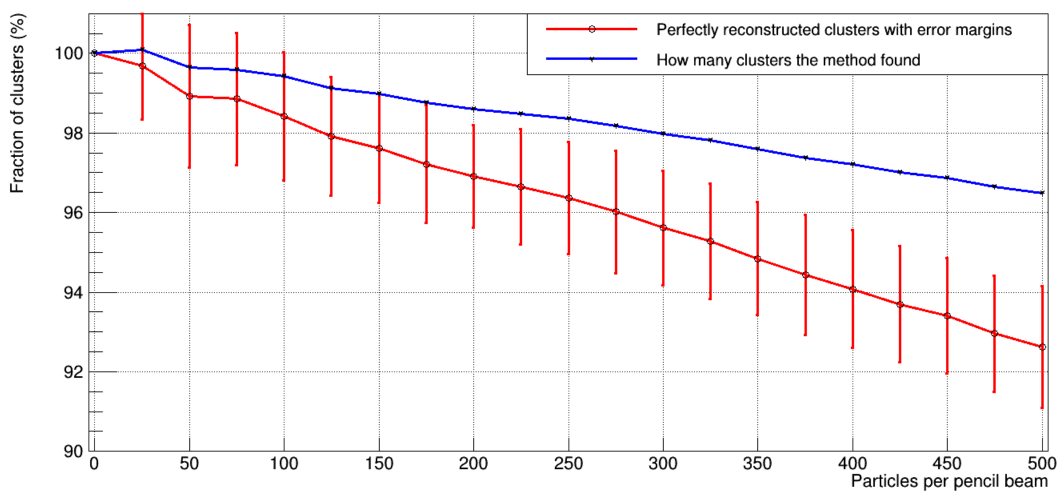



**Figure 4.17:** The inactive pixels cut - Illustration of how many clusters the cut was able to find versus how many perfectly reconstructed clusters which was found with error margins.

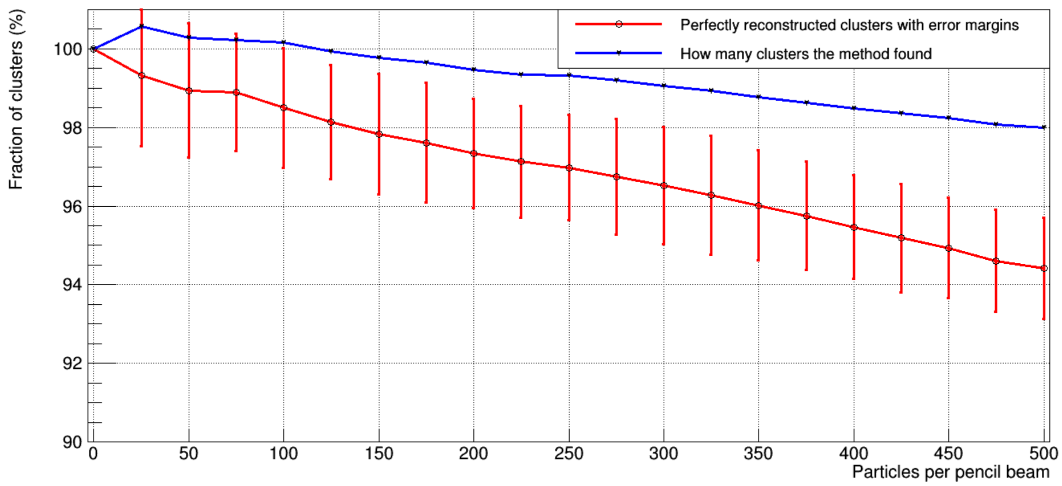



**Figure 4.18:** The inactive pixels and the fill percentage cuts - Illustration of how many clusters the cut was able to find versus how many perfectly reconstructed clusters which was found with error margins.

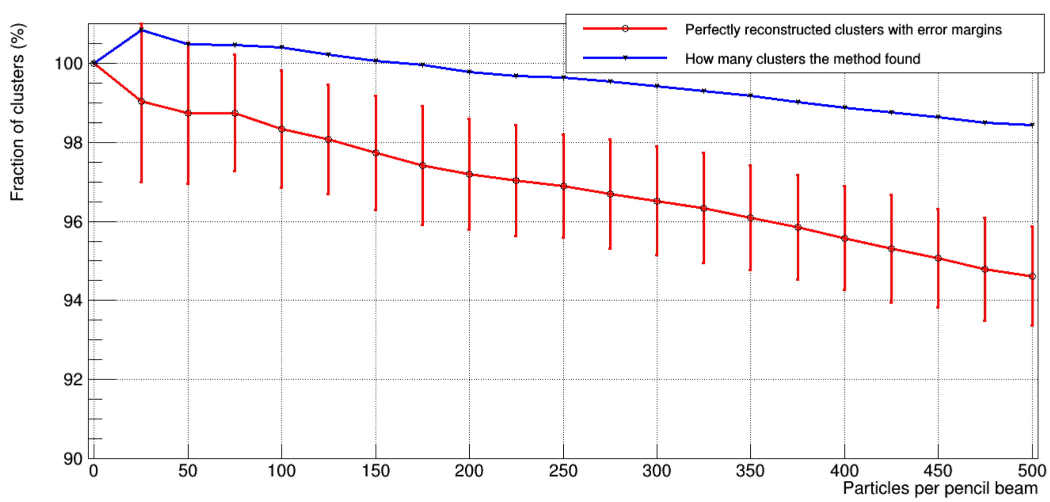



**Figure 4.19:** The asymmetric and the fill percentage cuts - Illustration of how many clusters the cut was able to find versus how many perfectly reconstructed clusters which was found with error margins.

# Chapter 5

# Conclusion and future work

This chapter contains a summary of the achieved results and further work based on the findings of the thesis.

## 5.1 Conclusion

The objective of this project was to simulate particle hits on an ALPIDE chip using a realistic data collection of pixel cluster shapes and then use different approaches to reconstruct the clusters. The challenge when reconstructing the clusters was to find a satisfying approach to separate the overlapping clusters. The reconstruction was done by experimenting with various cuts, which had different ways to distinguish abnormal clusters.

The basic method was a template for all the cuts when reconstructing clusters. 3 different cuts were tested: fill percentage cut, asymmetric cut and inactive pixels cut. The different cuts had various parameters to optimise in order to improve the percentage of perfectly reconstructed clusters. The identified clusters were additionally classified as two clusters if they exceeded the barrier (separator) values for height, width and size. All the simulations contained a beam size corresponding to a circular Gaussian with a standard deviation of 4 mm, and an average of 100-200 simulation per step in the graph to obtain a reasonably good statistical significance.

The optimal cut, based on the cuts tested in this thesis, was divided into 3 parts. These include the optimal solution for 0-100 clusters, 100-300 clusters and 300-500 clusters, respectively, in the detector. The inactive pixels cut had the most perfectly reconstructed clusters in the first part. The second and the third parts had 4 combinations of cuts which performed quite similarly, all different combinations containing the fill percentage square cut. Since all the 4 combinations of cuts maintained the settings from the fill percentage parameters and were not tuned individually, the optimal solution could differ slightly. However, there were some combinations which performed slightly better then the others in each part. The optimal cut for the second part, based on the selected parameters, was the inactive pixels cut combined with fill percentage cut. The third optimal cut, for 300-500, was the asymmetric cut combined with the fill percentage cut. The third optimal cut could have up to 350 clusters in the detector to get at least 96% perfectly reconstructed clusters. To obtain at least 98%, the maximum number of clusters is 125.

The different cuts developed in this thesis will result in a variation of 200-350 clusters in the detector to obtain at least 96% perfectly reconstructed clusters. Even 200 clusters in the detector will give a higher efficiency than the track reconstruction conducted in [6] by the Bergen proton CT research team.

## 5.2 Future work

The program which was developed in the thesis is not a final solution for the cluster reconstruction part of proton CT, only the beginning. Several extensions could be done to this simulation as further work:

- Cluster collection: The last method for separating clusters (section 3.1.2) could have been used on all the removed double clusters to get a larger cluster collection.
- Optimise all the parameter for each cut, beam size, particle thoroughly.
- Add noise and dead pixels in the detector and modify the basic method or the cuts to handle it.
- Apply cuts to multiple layers in path reconstruction part of proton CT.
- Develop other cuts to detect abnormal clusters.

# Bibliography


[1] M. Varga-Kofarago for the Bergen proton CT collaboration. "Medical Applications of the ALPIDE Detector". In: *MDPI* (May 2019). URL: `https://doi.org/10.3390/universe5050128`.

[2] Helge Egil Seime Pettersen. "A Digital Tracking Calorimeter for Proton Computed Tomography". PhD thesis. University of Bergen, Norway, 2018.

[3] Helge Egil Seime Pettersen et al. "Proton tracking in a high-granularity Digital Tracking Calorimeter for proton CT purposes". In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 860 (2017), pp. 51–61.

[4] Gianluca Aglieri Rinella, ALICE Collaboration, et al. "The ALPIDE pixel sensor chip for the upgrade of the ALICE Inner Tracking System". In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 845 (2017), pp. 583–587.

[5] Lukasz Maczewski. "Measurements and simulations of MAPS (Monolithic Active Pixel Sensors) response to charged particles-a study towards a vertex detector at the ILC". In: *arXiv preprint arXiv:1005.3710* (2010).

[6] Helge Egil Seime Pettersen, Silje Grimstad et al. *Design Optimization of a Pixel Based Range Telescope for Proton Computed Tomography.* Physica Medica, no. Special Issue: Advances in Geant4 for medicine in press.

[7] Norsk Helseinformatikk. *Kreft.* 2018. URL: `https://nhi.no/kroppen-var/sykdomsprosesser/kreft/` (visited on 01/18/2018).

[8] PTCOG Group. *Particle therapy facilities in operation.* 2019. URL: `https://www.ptcog.ch/index.php/facilities-in-operation` (visited on 01/27/2019).

[9] ProTom International. *Proton Therapy Treatment.* URL: `https://www.protominternational.com/proton-therapy/proton-therapy-treatment/` (visited on 01/29/2019).

[10] Einar Waldeland. "Proton therapy in Norway?" In: *Tidsskrift for den Norske laegeforening: tidsskrift for praktisk medicin, ny raekke* 130.8 (2010), p. 850. (Visited on 01/31/2019).

[11] Edward H. Shortliffe and James J. Cimino. *Biomedical informatics.* Forth. Springer, 2014. ISBN: 978-1-4471-4474-8. DOI: `10.1007/978-1-4471-4474-8`.

[12] Dave Fornell. *Imaging Considerations for Proton Therapy Treatment Planning.* 2016. URL: `https://www.itnonline.com/article/imaging-considerations-proton-therapy-treatment-planning` (visited on 01/27/2019).

[13] Rene Brun and Fons Rademakers. *ROOT — An Object Oriented Data Analysis Framework.* In: New Computing Techniques in Physics Research V 389, no. 1 (April 11, 1997): 81–86. URL: `https://doi.org/10.1016/S0168-9002(97)00048-X,%20https://root.cern.ch/`.

[14] Simon Kristian Huiberts. "Characterization of the ALPIDE chip with Helium-4 ions for Proton Computed Tomography". MA thesis. MSC thesis. University of Bergen, Norway, 2018. URL: `http://bora.uib.no/handle/1956/18748`.

[15] Ganesh Tambave et al. *Characterization of Monolithic CMOS Pixel Sensor Chip with Ion Beams for Application in Particle Computed Tomography*. Submitted to NIMA Proceedings.

[16] Christopher Z. Mooney. *Monte Carlo simulation*. Vol. no. 07-116. Sage university papers series. Thousand Oaks, CA: Sage Publications, 1997. ISBN: 1-4522-0986-3.

# Appendix A

| 0 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | Fill p. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 98.170 | 98.180 | 98.180 | 98.180 | 98.180 | 98.180 | 98.180 | 98.190 | 98.160 | 98.160 | 98.160 | 98.140 | 40 |
| 98.180 | 98.190 | 98.190 | 98.190 | 98.190 | 98.190 | 98.190 | 98.200 | 98.170 | 98.170 | 98.170 | 98.150 | 41 |
| 98.160 | 98.170 | 98.170 | 98.170 | 98.170 | 98.170 | 98.170 | 98.180 | 98.150 | 98.170 | 98.170 | 98.150 | 42 |
| 98.200 | 98.210 | 98.210 | 98.210 | 98.210 | 98.210 | 98.210 | 98.220 | 98.190 | 98.210 | 98.210 | 98.190 | 43 |
| 98.150 | 98.210 | 98.210 | 98.210 | 98.210 | 98.210 | 98.210 | 98.220 | 98.190 | 98.210 | 98.210 | 98.190 | 44 |
| 97.660 | 98.380 | 98.380 | 98.380 | 98.380 | 98.410 | 98.410 | 98.420 | 98.390 | 98.410 | 98.290 | 98.270 | 45 |
| 97.680 | 98.400 | 98.400 | 98.400 | 98.400 | 98.430 | 98.430 | 98.440 | 98.410 | 98.430 | 98.310 | 98.290 | 46 |
| 97.680 | 98.400 | 98.400 | 98.400 | 98.400 | 98.430 | 98.430 | 98.440 | 98.410 | 98.430 | 98.310 | 98.290 | 47 |
| 97.720 | 98.440 | 98.440 | 98.440 | 98.440 | 98.470 | 98.470 | 98.480 | 98.450 | 98.470 | 98.350 | 98.290 | 48 |
| 97.680 | 98.400 | 98.400 | 98.400 | 98.400 | 98.430 | 98.510 | **98.520** | 98.490 | 98.510 | 98.390 | 98.330 | 49 |
| 97.680 | 98.400 | 98.400 | 98.400 | 98.400 | 98.430 | 98.510 | **98.520** | 98.490 | 98.510 | 98.390 | 98.330 | 50 |
| 93.490 | 94.610 | 98.350 | 98.350 | 98.350 | 98.380 | 98.460 | 98.470 | 98.440 | 98.460 | 98.340 | 98.280 | 51 |
| 93.510 | 94.630 | 98.370 | 98.370 | 98.370 | 98.400 | 98.480 | 98.490 | 98.460 | 98.480 | 98.360 | 98.300 | 52 |
| 93.310 | 94.430 | 98.170 | 98.170 | 98.170 | 98.200 | 98.280 | 98.500 | 98.470 | 98.490 | 98.370 | 98.310 | 53 |
| 93.310 | 94.430 | 98.170 | 98.170 | 98.170 | 98.200 | 98.280 | 98.500 | 98.470 | 98.490 | 98.370 | 98.310 | 54 |
| 93.310 | 94.430 | 98.170 | 98.170 | 98.170 | 98.200 | 98.280 | 98.500 | 98.470 | 98.490 | 98.370 | 98.310 | 55 |
| 90.620 | 94.290 | 98.030 | 98.030 | 98.030 | 98.060 | 98.140 | 98.360 | 98.330 | 98.350 | 98.230 | 98.170 | 56 |
| 81.080 | 84.750 | 88.490 | 97.520 | 97.520 | 97.550 | 97.630 | 97.850 | 98.330 | 98.350 | 98.230 | 98.170 | 57 |
| 81.080 | 84.750 | 88.490 | 97.520 | 97.520 | 97.550 | 97.630 | 97.850 | 98.330 | 98.350 | 98.230 | 98.170 | 58 |
| 81.010 | 84.680 | 88.420 | 97.450 | 97.450 | 97.480 | 97.560 | 97.780 | 98.260 | 98.280 | 98.160 | 98.100 | 59 |
| 81.010 | 84.680 | 88.420 | 97.450 | 97.450 | 97.480 | 97.560 | 97.780 | 98.260 | 98.280 | 98.160 | 98.100 | 60 |
| 79.950 | 83.620 | 87.360 | 96.390 | 96.390 | 96.420 | 96.500 | 96.720 | 97.200 | 98.280 | 98.160 | 98.100 | 61 |
| 79.920 | 83.590 | 87.330 | 96.360 | 96.360 | 96.390 | 96.470 | 96.690 | 97.170 | 98.250 | 98.130 | 98.070 | 62 |
| 66.100 | 69.770 | 73.510 | 82.540 | 96.360 | 96.390 | 96.470 | 96.690 | 97.170 | 98.250 | 98.130 | 98.070 | 63 |
| 66.090 | 69.760 | 73.500 | 82.530 | 96.350 | 96.380 | 96.460 | 96.680 | 97.160 | 98.240 | 98.120 | 98.060 | 64 |
| 65.050 | 68.720 | 72.460 | 81.490 | 95.310 | 95.340 | 95.420 | 95.640 | 96.120 | 97.200 | 98.120 | 98.060 | 65 |
| 65.050 | 68.720 | 72.460 | 81.490 | 95.310 | 95.340 | 95.420 | 95.640 | 96.120 | 97.200 | 98.120 | 98.060 | 66 |
| 60.850 | 68.680 | 72.420 | 81.450 | 95.270 | 95.300 | 95.380 | 95.600 | 96.080 | 97.160 | 98.080 | 98.020 | 67 |
| 60.850 | 68.680 | 72.420 | 81.450 | 95.270 | 95.300 | 95.380 | 95.600 | 96.080 | 97.160 | 98.080 | 98.020 | 68 |
| 52.600 | 60.430 | 64.170 | 73.200 | 87.020 | 94.770 | 94.850 | 95.070 | 95.550 | 96.630 | 97.550 | 98.020 | 69 |
| 52.580 | 60.410 | 64.150 | 73.180 | 87.000 | 94.750 | 94.830 | 95.050 | 95.530 | 96.610 | 97.530 | 98.000 | 70 |

**Table 1:** Results from combining different variants of minimum sizes and fill percentage square approach. The simulation uses 100 cluster in the detector and beam size 4mm, which results in the highest perfectly reconstructed cluster to be 97.265%. There are 2 combination which results in the same highest score, both with minimum size 13, and fill percentage of 49% and 50%. Each value is an average of 100 simulations.

| 0 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | Fill p. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 97.320 | 97.327 | 97.327 | 97.313 | 97.313 | 97.313 | 97.313 | 97.320 | 97.307 | 97.307 | 97.307 | 97.293 | 40 |
| 97.407 | 97.413 | 97.413 | 97.400 | 97.400 | 97.400 | 97.400 | 97.407 | 97.393 | 97.393 | 97.393 | 97.380 | 41 |
| 97.393 | 97.400 | 97.400 | 97.387 | 97.387 | 97.387 | 97.387 | 97.393 | 97.380 | 97.393 | 97.393 | 97.380 | 42 |
| 97.493 | 97.500 | 97.500 | 97.487 | 97.487 | 97.487 | 97.487 | 97.493 | 97.480 | 97.493 | 97.493 | 97.480 | 43 |
| 97.460 | 97.500 | 97.500 | 97.487 | 97.487 | 97.487 | 97.487 | 97.493 | 97.480 | 97.493 | 97.493 | 97.480 | 44 |
| 96.967 | 97.627 | 97.627 | 97.613 | 97.613 | 97.627 | 97.627 | 97.633 | 97.620 | 97.633 | 97.547 | 97.533 | 45 |
| 96.980 | 97.640 | 97.640 | 97.627 | 97.627 | 97.640 | 97.640 | 97.647 | 97.633 | 97.647 | 97.560 | 97.547 | 46 |
| 97.000 | 97.660 | 97.660 | 97.647 | 97.647 | 97.660 | 97.660 | 97.667 | 97.653 | 97.667 | 97.580 | 97.567 | 47 |
| 97.053 | 97.713 | 97.713 | 97.700 | 97.700 | 97.713 | 97.713 | 97.720 | 97.707 | 97.720 | 97.633 | 97.567 | 48 |
| 97.047 | 97.707 | 97.707 | 97.693 | 97.693 | 97.707 | 97.780 | 97.787 | 97.773 | 97.787 | 97.700 | 97.633 | 49 |
| 97.047 | 97.707 | 97.707 | 97.693 | 97.693 | 97.707 | 97.780 | 97.787 | 97.773 | 97.787 | 97.700 | 97.633 | 50 |
| 92.987 | 94.087 | 97.720 | 97.707 | 97.707 | 97.720 | 97.793 | **97.800** | 97.787 | **97.800** | 97.713 | 97.647 | 51 |
| 92.987 | 94.087 | 97.720 | 97.707 | 97.707 | 97.720 | 97.793 | **97.800** | 97.787 | **97.800** | 97.713 | 97.647 | 52 |
| 92.827 | 93.927 | 97.560 | 97.547 | 97.547 | 97.560 | 97.633 | **97.800** | 97.787 | **97.800** | 97.713 | 97.647 | 53 |
| 92.827 | 93.927 | 97.560 | 97.547 | 97.547 | 97.560 | 97.633 | **97.800** | 97.787 | **97.800** | 97.713 | 97.647 | 54 |
| 92.827 | 93.927 | 97.560 | 97.547 | 97.547 | 97.560 | 97.633 | **97.800** | 97.787 | **97.800** | 97.713 | 97.647 | 55 |
| 90.353 | 93.867 | 97.500 | 97.487 | 97.487 | 97.500 | 97.573 | 97.740 | 97.727 | 97.740 | 97.653 | 97.587 | 56 |
| 80.920 | 84.433 | 88.067 | 97.107 | 97.107 | 97.120 | 97.193 | 97.360 | 97.727 | 97.740 | 97.653 | 97.587 | 57 |
| 80.920 | 84.433 | 88.067 | 97.107 | 97.107 | 97.120 | 97.193 | 97.360 | 97.727 | 97.740 | 97.653 | 97.587 | 58 |
| 80.880 | 84.393 | 88.027 | 97.067 | 97.067 | 97.080 | 97.153 | 97.320 | 97.687 | 97.700 | 97.613 | 97.547 | 59 |
| 80.880 | 84.393 | 88.027 | 97.067 | 97.067 | 97.080 | 97.153 | 97.320 | 97.687 | 97.700 | 97.613 | 97.547 | 60 |
| 79.933 | 83.447 | 87.080 | 96.120 | 96.120 | 96.133 | 96.207 | 96.373 | 96.740 | 97.700 | 97.613 | 97.547 | 61 |
| 79.913 | 83.427 | 87.060 | 96.100 | 96.100 | 96.113 | 96.187 | 96.353 | 96.720 | 97.680 | 97.593 | 97.527 | 62 |
| 66.393 | 69.907 | 73.540 | 82.580 | 96.100 | 96.113 | 96.187 | 96.353 | 96.720 | 97.680 | 97.593 | 97.527 | 63 |
| 66.400 | 69.913 | 73.547 | 82.587 | 96.107 | 96.120 | 96.193 | 96.360 | 96.727 | 97.687 | 97.600 | 97.533 | 64 |
| 65.360 | 68.873 | 72.507 | 81.547 | 95.067 | 95.080 | 95.153 | 95.320 | 95.687 | 96.647 | 97.600 | 97.533 | 65 |
| 65.360 | 68.873 | 72.507 | 81.547 | 95.067 | 95.080 | 95.153 | 95.320 | 95.687 | 96.647 | 97.600 | 97.533 | 66 |
| 61.160 | 68.827 | 72.460 | 81.500 | 95.020 | 95.033 | 95.107 | 95.273 | 95.640 | 96.600 | 97.553 | 97.487 | 67 |
| 61.160 | 68.827 | 72.460 | 81.500 | 95.020 | 95.033 | 95.107 | 95.273 | 95.640 | 96.600 | 97.553 | 97.487 | 68 |
| 52.953 | 60.620 | 64.253 | 73.293 | 86.813 | 94.393 | 94.467 | 94.633 | 95.000 | 95.960 | 96.913 | 97.487 | 69 |
| 52.967 | 60.633 | 64.267 | 73.307 | 86.827 | 94.407 | 94.480 | 94.647 | 95.013 | 95.973 | 96.927 | 97.500 | 70 |

**Table 2:** Results from combining different variants of minimum sizes and fill percentage square approach. The simulation uses 150 cluster in the detector and beam size 4mm, which results in the highest perfectly reconstructed cluster to be 97.8%. There are 10 combination which results in the same highest score, including fill percentage 55% and minimum size 13. Each value is an average of 100 simulations.

| 0 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | Fill p. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 96.390 | 96.400 | 96.400 | 96.395 | 96.395 | 96.395 | 96.395 | 96.405 | 96.385 | 96.385 | 96.355 | 96.325 | 40 |
| 96.505 | 96.515 | 96.515 | 96.510 | 96.510 | 96.510 | 96.510 | 96.520 | 96.500 | 96.500 | 96.470 | 96.440 | 41 |
| 96.525 | 96.535 | 96.535 | 96.530 | 96.530 | 96.530 | 96.530 | 96.540 | 96.520 | 96.500 | 96.470 | 96.440 | 42 |
| 96.635 | 96.645 | 96.645 | 96.640 | 96.640 | 96.640 | 96.640 | 96.650 | 96.630 | 96.610 | 96.580 | 96.550 | 43 |
| 96.635 | 96.675 | 96.675 | 96.670 | 96.670 | 96.670 | 96.670 | 96.680 | 96.660 | 96.640 | 96.610 | 96.580 | 44 |
| 96.225 | 96.840 | 96.840 | 96.835 | 96.835 | 96.855 | 96.855 | 96.865 | 96.845 | 96.825 | 96.700 | 96.670 | 45 |
| 96.255 | 96.870 | 96.870 | 96.865 | 96.865 | 96.885 | 96.885 | 96.895 | 96.875 | 96.855 | 96.730 | 96.700 | 46 |
| 96.280 | 96.895 | 96.895 | 96.890 | 96.890 | 96.910 | 96.910 | 96.920 | 96.900 | 96.880 | 96.755 | 96.725 | 47 |
| 96.400 | 97.015 | 97.015 | 97.010 | 97.010 | 97.030 | 97.030 | 97.040 | 97.020 | 97.000 | 96.875 | 96.725 | 48 |
| 96.405 | 97.020 | 97.020 | 97.015 | 97.015 | 97.035 | 97.110 | 97.120 | 97.100 | 97.080 | 96.955 | 96.805 | 49 |
| 96.405 | 97.020 | 97.020 | 97.015 | 97.015 | 97.035 | 97.110 | 97.120 | 97.100 | 97.080 | 96.955 | 96.805 | 50 |
| 92.470 | 93.540 | 97.080 | 97.075 | 97.075 | 97.095 | 97.170 | 97.180 | 97.160 | 97.140 | 97.015 | 96.865 | 51 |
| 92.480 | 93.550 | 97.090 | 97.085 | 97.085 | 97.105 | 97.180 | 97.190 | 97.170 | 97.150 | 97.025 | 96.875 | 52 |
| 92.410 | 93.480 | 97.020 | 97.015 | 97.015 | 97.035 | 97.110 | 97.250 | 97.230 | 97.210 | 97.085 | 96.935 | 53 |
| 92.425 | 93.495 | 97.035 | 97.030 | 97.030 | 97.050 | 97.125 | **97.265** | 97.245 | 97.225 | 97.100 | 96.950 | 54 |
| 92.425 | 93.495 | 97.035 | 97.030 | 97.030 | 97.050 | 97.125 | **97.265** | 97.245 | 97.225 | 97.100 | 96.950 | 55 |
| 90.085 | 93.470 | 97.010 | 97.005 | 97.005 | 97.025 | 97.100 | 97.240 | 97.220 | 97.200 | 97.075 | 96.925 | 56 |
| 80.655 | 84.040 | 87.580 | 96.740 | 96.740 | 96.760 | 96.835 | 96.975 | 97.220 | 97.200 | 97.075 | 96.925 | 57 |
| 80.655 | 84.040 | 87.580 | 96.740 | 96.740 | 96.760 | 96.835 | 96.975 | 97.220 | 97.200 | 97.075 | 96.925 | 58 |
| 80.630 | 84.015 | 87.555 | 96.715 | 96.715 | 96.735 | 96.810 | 96.950 | 97.195 | 97.175 | 97.050 | 96.900 | 59 |
| 80.640 | 84.025 | 87.565 | 96.725 | 96.725 | 96.745 | 96.820 | 96.960 | 97.205 | 97.185 | 97.060 | 96.910 | 60 |
| 79.720 | 83.105 | 86.645 | 95.805 | 95.805 | 95.825 | 95.900 | 96.040 | 96.285 | 97.185 | 97.060 | 96.910 | 61 |
| 79.705 | 83.090 | 86.630 | 95.790 | 95.790 | 95.810 | 95.885 | 96.025 | 96.270 | 97.170 | 97.045 | 96.895 | 62 |
| 66.360 | 69.745 | 73.285 | 82.445 | 95.790 | 95.810 | 95.885 | 96.025 | 96.270 | 97.170 | 97.045 | 96.895 | 63 |
| 66.385 | 69.770 | 73.310 | 82.470 | 95.815 | 95.835 | 95.910 | 96.050 | 96.295 | 97.195 | 97.070 | 96.920 | 64 |
| 65.375 | 68.760 | 72.300 | 81.460 | 94.805 | 94.825 | 94.900 | 95.040 | 95.285 | 96.185 | 97.070 | 96.920 | 65 |
| 65.375 | 68.760 | 72.300 | 81.460 | 94.805 | 94.825 | 94.900 | 95.040 | 95.285 | 96.185 | 97.070 | 96.920 | 66 |
| 61.400 | 68.725 | 72.265 | 81.425 | 94.770 | 94.790 | 94.865 | 95.005 | 95.250 | 96.150 | 97.035 | 96.885 | 67 |
| 61.400 | 68.725 | 72.265 | 81.425 | 94.770 | 94.790 | 94.865 | 95.005 | 95.250 | 96.150 | 97.035 | 96.885 | 68 |
| 53.100 | 60.425 | 63.965 | 73.125 | 86.470 | 94.185 | 94.260 | 94.400 | 94.645 | 95.545 | 96.430 | 96.885 | 69 |
| 53.105 | 60.430 | 63.970 | 73.130 | 86.475 | 94.190 | 94.265 | 94.405 | 94.650 | 95.550 | 96.435 | 96.890 | 70 |

**Table 3:** Results from combining different variants of minimum sizes and fill percentage square approach. The simulation uses 200 cluster in the detector and beam size 4mm, which results in the highest perfectly reconstructed cluster to be 97.265%. There are 2 combination which results in the same highest score, including fill percentage 55% and minimum size 13. Each value is an average of 100 simulations.

| 0 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | Fill p. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 95.716 | 95.728 | 95.728 | 95.724 | 95.724 | 95.724 | 95.724 | 95.732 | 95.684 | 95.684 | 95.660 | 95.620 | 40 |
| 95.872 | 95.884 | 95.884 | 95.880 | 95.880 | 95.880 | 95.880 | 95.888 | 95.840 | 95.840 | 95.816 | 95.776 | 41 |
| 95.896 | 95.908 | 95.908 | 95.904 | 95.904 | 95.904 | 95.904 | 95.912 | 95.864 | 95.840 | 95.816 | 95.776 | 42 |
| 96.040 | 96.052 | 96.052 | 96.048 | 96.048 | 96.048 | 96.048 | 96.056 | 96.008 | 95.984 | 95.960 | 95.920 | 43 |
| 96.040 | 96.084 | 96.084 | 96.080 | 96.080 | 96.080 | 96.080 | 96.088 | 96.040 | 96.016 | 95.992 | 95.952 | 44 |
| 95.720 | 96.336 | 96.336 | 96.332 | 96.332 | 96.340 | 96.340 | 96.348 | 96.300 | 96.276 | 96.120 | 96.080 | 45 |
| 95.744 | 96.360 | 96.360 | 96.356 | 96.356 | 96.364 | 96.364 | 96.372 | 96.324 | 96.300 | 96.144 | 96.104 | 46 |
| 95.792 | 96.408 | 96.408 | 96.404 | 96.404 | 96.412 | 96.412 | 96.420 | 96.372 | 96.348 | 96.192 | 96.152 | 47 |
| 95.960 | 96.576 | 96.576 | 96.572 | 96.572 | 96.580 | 96.580 | 96.588 | 96.540 | 96.516 | 96.360 | 96.152 | 48 |
| 95.956 | 96.572 | 96.572 | 96.568 | 96.568 | 96.576 | 96.660 | 96.668 | 96.620 | 96.596 | 96.440 | 96.232 | 49 |
| 95.956 | 96.572 | 96.572 | 96.568 | 96.568 | 96.576 | 96.660 | 96.668 | 96.620 | 96.596 | 96.440 | 96.232 | 50 |
| 92.012 | 93.096 | 96.636 | 96.632 | 96.632 | 96.640 | 96.724 | 96.732 | 96.684 | 96.660 | 96.504 | 96.296 | 51 |
| 92.052 | 93.136 | 96.676 | 96.672 | 96.672 | 96.680 | 96.764 | 96.772 | 96.724 | 96.700 | 96.544 | 96.336 | 52 |
| 92.032 | 93.116 | 96.656 | 96.652 | 96.652 | 96.660 | 96.744 | 96.880 | 96.832 | 96.808 | 96.652 | 96.444 | 53 |
| 92.048 | 93.132 | 96.672 | 96.668 | 96.668 | 96.676 | 96.760 | **96.896** | 96.848 | 96.824 | 96.668 | 96.460 | 54 |
| 92.048 | 93.132 | 96.672 | 96.668 | 96.668 | 96.676 | 96.760 | **96.896** | 96.848 | 96.824 | 96.668 | 96.460 | 55 |
| 89.724 | 93.116 | 96.656 | 96.652 | 96.652 | 96.660 | 96.744 | 96.880 | 96.832 | 96.808 | 96.652 | 96.444 | 56 |
| 80.360 | 83.752 | 87.292 | 96.404 | 96.404 | 96.412 | 96.496 | 96.632 | 96.832 | 96.808 | 96.652 | 96.444 | 57 |
| 80.360 | 83.752 | 87.292 | 96.404 | 96.404 | 96.412 | 96.496 | 96.632 | 96.832 | 96.808 | 96.652 | 96.444 | 58 |
| 80.356 | 83.748 | 87.288 | 96.400 | 96.400 | 96.408 | 96.492 | 96.628 | 96.828 | 96.804 | 96.648 | 96.440 | 59 |
| 80.372 | 83.764 | 87.304 | 96.416 | 96.416 | 96.424 | 96.508 | 96.644 | 96.844 | 96.820 | 96.664 | 96.456 | 60 |
| 79.504 | 82.896 | 86.436 | 95.548 | 95.548 | 95.556 | 95.640 | 95.776 | 95.976 | 96.820 | 96.664 | 96.456 | 61 |
| 79.508 | 82.900 | 86.440 | 95.552 | 95.552 | 95.560 | 95.644 | 95.780 | 95.980 | 96.824 | 96.668 | 96.460 | 62 |
| 66.296 | 69.688 | 73.228 | 82.340 | 95.552 | 95.560 | 95.644 | 95.780 | 95.980 | 96.824 | 96.668 | 96.460 | 63 |
| 66.316 | 69.708 | 73.248 | 82.360 | 95.572 | 95.580 | 95.664 | 95.800 | 96.000 | 96.844 | 96.688 | 96.480 | 64 |
| 65.372 | 68.764 | 72.304 | 81.416 | 94.628 | 94.636 | 94.720 | 94.856 | 95.056 | 95.900 | 96.688 | 96.480 | 65 |
| 65.372 | 68.764 | 72.304 | 81.416 | 94.628 | 94.636 | 94.720 | 94.856 | 95.056 | 95.900 | 96.688 | 96.480 | 66 |
| 61.360 | 68.736 | 72.276 | 81.388 | 94.600 | 94.608 | 94.692 | 94.828 | 95.028 | 95.872 | 96.660 | 96.452 | 67 |
| 61.360 | 68.736 | 72.276 | 81.388 | 94.600 | 94.608 | 94.692 | 94.828 | 95.028 | 95.872 | 96.660 | 96.452 | 68 |
| 53.256 | 60.632 | 64.172 | 73.284 | 86.496 | 94.092 | 94.176 | 94.312 | 94.512 | 95.356 | 96.144 | 96.452 | 69 |
| 53.276 | 60.652 | 64.192 | 73.304 | 86.516 | 94.112 | 94.196 | 94.332 | 94.532 | 95.376 | 96.164 | 96.472 | 70 |

**Table 4:** Results from combining different variants of minimum sizes and fill percentage square approach. The simulation uses 250 cluster in the detector and beam size 4mm, which results in the highest perfectly reconstructed cluster to be 96.896%. There are 2 combination which results in the same highest score, including fill percentage 55% and minimum size 13. Each value is an average of 100 simulations.

| 0 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | Fill p. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 94.787 | 94.800 | 94.800 | 94.790 | 94.790 | 94.790 | 94.790 | 94.800 | 94.747 | 94.733 | 94.707 | 94.647 | 40 |
| 94.990 | 95.003 | 95.003 | 94.993 | 94.993 | 94.993 | 94.993 | 95.003 | 94.950 | 94.937 | 94.910 | 94.850 | 41 |
| 95.050 | 95.063 | 95.063 | 95.053 | 95.053 | 95.053 | 95.053 | 95.063 | 95.010 | 94.937 | 94.910 | 94.850 | 42 |
| 95.243 | 95.257 | 95.257 | 95.247 | 95.247 | 95.247 | 95.247 | 95.257 | 95.203 | 95.130 | 95.103 | 95.043 | 43 |
| 95.260 | 95.303 | 95.303 | 95.293 | 95.293 | 95.293 | 95.293 | 95.303 | 95.250 | 95.177 | 95.150 | 95.090 | 44 |
| 95.033 | 95.657 | 95.657 | 95.647 | 95.647 | 95.647 | 95.647 | 95.657 | 95.603 | 95.530 | 95.310 | 95.250 | 45 |
| 95.053 | 95.677 | 95.677 | 95.667 | 95.667 | 95.667 | 95.667 | 95.677 | 95.623 | 95.550 | 95.330 | 95.270 | 46 |
| 95.110 | 95.733 | 95.733 | 95.723 | 95.723 | 95.723 | 95.723 | 95.733 | 95.680 | 95.607 | 95.387 | 95.327 | 47 |
| 95.333 | 95.957 | 95.957 | 95.947 | 95.947 | 95.947 | 95.947 | 95.957 | 95.903 | 95.830 | 95.610 | 95.327 | 48 |
| 95.340 | 95.963 | 95.963 | 95.953 | 95.953 | 95.953 | 96.040 | 96.050 | 95.997 | 95.923 | 95.703 | 95.420 | 49 |
| 95.340 | 95.963 | 95.963 | 95.953 | 95.953 | 95.953 | 96.040 | 96.050 | 95.997 | 95.923 | 95.703 | 95.420 | 50 |
| 91.510 | 92.587 | 96.133 | 96.123 | 96.123 | 96.123 | 96.210 | 96.220 | 96.167 | 96.093 | 95.873 | 95.590 | 51 |
| 91.543 | 92.620 | 96.167 | 96.157 | 96.157 | 96.157 | 96.243 | 96.253 | 96.200 | 96.127 | 95.907 | 95.623 | 52 |
| 91.567 | 92.643 | 96.190 | 96.180 | 96.180 | 96.180 | 96.267 | 96.393 | 96.340 | 96.267 | 96.047 | 95.763 | 53 |
| 91.590 | 92.667 | 96.213 | 96.203 | 96.203 | 96.203 | 96.290 | 96.417 | 96.363 | 96.290 | 96.070 | 95.787 | 54 |
| 91.590 | 92.667 | 96.213 | 96.203 | 96.203 | 96.203 | 96.290 | 96.417 | 96.363 | 96.290 | 96.070 | 95.787 | 55 |
| 89.380 | 92.707 | 96.253 | 96.243 | 96.243 | 96.243 | 96.330 | **96.457** | 96.403 | 96.330 | 96.110 | 95.827 | 56 |
| 80.087 | 83.413 | 86.960 | 96.037 | 96.037 | 96.037 | 96.123 | 96.250 | 96.403 | 96.330 | 96.110 | 95.827 | 57 |
| 80.100 | 83.427 | 86.973 | 96.050 | 96.050 | 96.050 | 96.137 | 96.263 | 96.417 | 96.343 | 96.123 | 95.840 | 58 |
| 80.113 | 83.440 | 86.987 | 96.063 | 96.063 | 96.063 | 96.150 | 96.277 | 96.430 | 96.357 | 96.137 | 95.853 | 59 |
| 80.127 | 83.453 | 87.000 | 96.077 | 96.077 | 96.077 | 96.163 | 96.290 | 96.443 | 96.370 | 96.150 | 95.867 | 60 |
| 79.300 | 82.627 | 86.173 | 95.250 | 95.250 | 95.250 | 95.337 | 95.463 | 95.617 | 96.370 | 96.150 | 95.867 | 61 |
| 79.307 | 82.633 | 86.180 | 95.257 | 95.257 | 95.257 | 95.343 | 95.470 | 95.623 | 96.377 | 96.157 | 95.873 | 62 |
| 66.293 | 69.620 | 73.167 | 82.243 | 95.257 | 95.257 | 95.343 | 95.470 | 95.623 | 96.377 | 96.157 | 95.873 | 63 |
| 66.333 | 69.660 | 73.207 | 82.283 | 95.297 | 95.297 | 95.383 | 95.510 | 95.663 | 96.417 | 96.197 | 95.913 | 64 |
| 65.480 | 68.807 | 72.353 | 81.430 | 94.443 | 94.443 | 94.530 | 94.657 | 94.810 | 95.563 | 96.197 | 95.913 | 65 |
| 65.480 | 68.807 | 72.353 | 81.430 | 94.443 | 94.443 | 94.530 | 94.657 | 94.810 | 95.563 | 96.197 | 95.913 | 66 |
| 61.613 | 68.793 | 72.340 | 81.417 | 94.430 | 94.430 | 94.517 | 94.643 | 94.797 | 95.550 | 96.183 | 95.900 | 67 |
| 61.613 | 68.793 | 72.340 | 81.417 | 94.430 | 94.430 | 94.517 | 94.643 | 94.797 | 95.550 | 96.183 | 95.900 | 68 |
| 53.670 | 60.850 | 64.397 | 73.473 | 86.487 | 93.927 | 94.013 | 94.140 | 94.293 | 95.047 | 95.680 | 95.900 | 69 |
| 53.687 | 60.867 | 64.413 | 73.490 | 86.503 | 93.943 | 94.030 | 94.157 | 94.310 | 95.063 | 95.697 | 95.917 | 70 |

**Table 5:** Results from combining different variants of minimum sizes and fill percentage square approach. The simulation uses 300 cluster in the detector and beam size 4mm, which results in the highest perfectly reconstructed cluster to be 96.457%. The best combination was fill percentage 56% and minimum size 13. Each value is an average of 100 simulations.

# Appendix B

| 0 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | Fill p. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 98.000 | 98.000 | 98.000 | 98.000 | 98.000 | 98.000 | 97.980 | 97.980 | 97.980 | 97.980 | 97.980 | 97.980 | 55 |
| 97.440 | 98.020 | 98.020 | 98.020 | 98.020 | 98.000 | 97.980 | 97.980 | 97.980 | 97.980 | 97.980 | 97.980 | 56 |
| 97.420 | 98.000 | 98.000 | 98.090 | 98.090 | 98.070 | 98.050 | 98.050 | 98.050 | 98.050 | 98.050 | 98.000 | 57 |
| 97.520 | 98.100 | 98.100 | 98.190 | 98.190 | 98.170 | 98.150 | 98.150 | 98.150 | 98.150 | 98.150 | 98.100 | 58 |
| 97.500 | 98.080 | 98.080 | 98.170 | 98.170 | 98.150 | 98.130 | 98.130 | 98.130 | 98.130 | 98.130 | 98.080 | 59 |
| 97.520 | 98.100 | 98.100 | 98.190 | 98.190 | 98.170 | 98.150 | 98.150 | 98.150 | 98.150 | 98.150 | 98.100 | 60 |
| 97.440 | 98.020 | 98.020 | 98.110 | 98.110 | 98.090 | 98.150 | 98.150 | 98.150 | 98.160 | 98.160 | 98.110 | 61 |
| 97.440 | 98.020 | 98.020 | 98.110 | 98.110 | 98.090 | 98.150 | 98.150 | 98.150 | 98.160 | 98.160 | 98.110 | 62 |
| 96.180 | 96.760 | 96.760 | 96.850 | 98.100 | 98.080 | 98.140 | 98.140 | 98.140 | 98.170 | 98.170 | 98.120 | 63 |
| 96.200 | 96.780 | 96.780 | 96.870 | 98.120 | 98.100 | 98.160 | 98.160 | 98.160 | 98.190 | 98.190 | 98.140 | 64 |
| 96.200 | 96.780 | 96.780 | 96.870 | 98.120 | 98.100 | 98.160 | 98.160 | 98.160 | 98.190 | 98.230 | 98.180 | 65 |
| 96.030 | 96.610 | 96.610 | 96.700 | 97.950 | 97.930 | 97.990 | 98.210 | 98.210 | 98.240 | 98.280 | 98.230 | 66 |
| 90.660 | 92.910 | 96.650 | 96.740 | 97.990 | 97.970 | 98.030 | 98.250 | 98.250 | 98.280 | 98.230 | 98.180 | 67 |
| 90.680 | 92.930 | 96.670 | 96.760 | 98.010 | 97.990 | 98.050 | 98.270 | 98.270 | **98.300** | 98.250 | 98.200 | 68 |
| 85.610 | 87.860 | 91.600 | 91.690 | 92.940 | 97.840 | 97.900 | 98.120 | 98.120 | 98.150 | 98.100 | 98.200 | 69 |
| 85.590 | 87.840 | 91.580 | 91.670 | 92.920 | 97.820 | 97.880 | 98.100 | 98.100 | 98.130 | 98.080 | 98.180 | 70 |
| 85.140 | 87.390 | 91.130 | 91.220 | 92.470 | 97.370 | 97.430 | 97.650 | 98.050 | 98.080 | 98.030 | 98.140 | 71 |
| 85.150 | 87.400 | 91.140 | 91.230 | 92.480 | 97.380 | 97.440 | 97.660 | 98.060 | 98.090 | 98.040 | 98.150 | 72 |
| 84.920 | 87.170 | 90.910 | 91.000 | 92.250 | 97.150 | 97.210 | 97.440 | 97.840 | 97.870 | 97.820 | 97.930 | 73 |
| 84.840 | 87.090 | 90.830 | 90.920 | 92.170 | 97.120 | 97.180 | 97.410 | 97.810 | 97.840 | 97.790 | 97.900 | 74 |
| 84.860 | 87.110 | 90.850 | 90.940 | 92.190 | 97.140 | 97.200 | 97.430 | 97.830 | 97.860 | 97.810 | 97.920 | 75 |
| 64.660 | 67.780 | 71.520 | 80.550 | 81.800 | 86.750 | 96.110 | 96.340 | 96.740 | 97.820 | 97.770 | 97.880 | 76 |
| 64.460 | 67.580 | 71.320 | 80.350 | 81.600 | 86.550 | 95.910 | 96.140 | 96.540 | 97.620 | 97.570 | 97.680 | 77 |
| 57.120 | 67.610 | 71.350 | 80.380 | 81.630 | 86.580 | 95.940 | 96.170 | 96.540 | 97.620 | 97.570 | 97.680 | 78 |
| 57.180 | 67.670 | 71.410 | 80.440 | 81.690 | 86.640 | 96.000 | 96.230 | 96.600 | 97.680 | 97.630 | 97.740 | 79 |
| 57.170 | 67.660 | 71.400 | 80.430 | 81.680 | 86.630 | 95.990 | 96.220 | 96.590 | 97.670 | 97.620 | 97.730 | 80 |

**Table 6:** Results from combining different variants of minimum sizes and fill percentage rectangle approach. The simulation uses 100 cluster in the detector and beam size 4mm, which results in the highest perfectly reconstructed cluster to be 98.300%. The best score is a combination of fill percentage 68% and minimum size 15. Each value is an average of 100 simulations.

| 0 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | Fill p. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 97.160 | 97.160 | 97.160 | 97.160 | 97.160 | 97.160 | 97.147 | 97.133 | 97.133 | 97.133 | 97.133 | 97.133 | 55 |
| 96.680 | 97.200 | 97.200 | 97.200 | 97.200 | 97.187 | 97.173 | 97.160 | 97.160 | 97.160 | 97.160 | 97.160 | 56 |
| 96.647 | 97.167 | 97.167 | 97.253 | 97.253 | 97.240 | 97.227 | 97.213 | 97.200 | 97.200 | 97.200 | 97.173 | 57 |
| 96.767 | 97.287 | 97.287 | 97.373 | 97.373 | 97.360 | 97.347 | 97.333 | 97.320 | 97.320 | 97.320 | 97.293 | 58 |
| 96.747 | 97.267 | 97.267 | 97.353 | 97.353 | 97.340 | 97.327 | 97.313 | 97.300 | 97.300 | 97.300 | 97.273 | 59 |
| 96.747 | 97.267 | 97.267 | 97.353 | 97.353 | 97.340 | 97.327 | 97.313 | 97.300 | 97.300 | 97.300 | 97.273 | 60 |
| 96.773 | 97.293 | 97.293 | 97.367 | 97.367 | 97.353 | 97.400 | 97.387 | 97.373 | 97.380 | 97.380 | 97.353 | 61 |
| 96.780 | 97.300 | 97.300 | 97.373 | 97.373 | 97.360 | 97.407 | 97.393 | 97.380 | 97.387 | 97.387 | 97.360 | 62 |
| 95.553 | 96.073 | 96.073 | 96.147 | 97.353 | 97.340 | 97.387 | 97.373 | 97.360 | 97.393 | 97.393 | 97.367 | 63 |
| 95.587 | 96.107 | 96.107 | 96.180 | 97.387 | 97.373 | 97.420 | 97.407 | 97.393 | 97.427 | 97.427 | 97.400 | 64 |
| 95.613 | 96.133 | 96.133 | 96.207 | 97.413 | 97.400 | 97.447 | 97.433 | 97.420 | 97.453 | 97.467 | 97.440 | 65 |
| 95.480 | 96.000 | 96.000 | 96.073 | 97.280 | 97.267 | 97.313 | 97.467 | 97.453 | 97.487 | 97.500 | 97.473 | 66 |
| 90.207 | 92.427 | 96.060 | 96.133 | 97.340 | 97.327 | 97.373 | 97.527 | 97.513 | 97.547 | 97.487 | 97.460 | 67 |
| 90.233 | 92.453 | 96.087 | 96.160 | 97.367 | 97.353 | 97.400 | 97.553 | 97.540 | 97.573 | 97.513 | 97.487 | 68 |
| 85.213 | 87.433 | 91.067 | 91.140 | 92.347 | 97.213 | 97.260 | 97.413 | 97.400 | 97.433 | 97.373 | 97.540 | 69 |
| 85.227 | 87.447 | 91.080 | 91.153 | 92.360 | 97.227 | 97.273 | 97.427 | 97.413 | 97.447 | 97.387 | 97.553 | 70 |
| 84.913 | 87.133 | 90.767 | 90.840 | 92.047 | 96.913 | 96.960 | 97.113 | 97.413 | 97.447 | 97.387 | 97.527 | 71 |
| 84.967 | 87.187 | 90.820 | 90.893 | 92.100 | 96.967 | 97.013 | 97.167 | 97.467 | 97.500 | 97.440 | **97.580** | 72 |
| 84.740 | 86.960 | 90.593 | 90.667 | 91.873 | 96.740 | 96.787 | 96.947 | 97.247 | 97.280 | 97.220 | 97.360 | 73 |
| 84.673 | 86.893 | 90.527 | 90.600 | 91.807 | 96.700 | 96.747 | 96.907 | 97.207 | 97.240 | 97.180 | 97.320 | 74 |
| 84.687 | 86.907 | 90.540 | 90.613 | 91.820 | 96.713 | 96.760 | 96.920 | 97.220 | 97.253 | 97.193 | 97.333 | 75 |
| 64.633 | 67.720 | 71.353 | 80.393 | 81.600 | 86.493 | 95.827 | 95.987 | 96.287 | 97.280 | 97.220 | 97.360 | 76 |
| 64.433 | 67.520 | 71.153 | 80.193 | 81.400 | 86.293 | 95.627 | 95.787 | 96.087 | 97.080 | 97.020 | 97.160 | 77 |
| 57.107 | 67.533 | 71.167 | 80.207 | 81.413 | 86.307 | 95.640 | 95.800 | 96.087 | 97.080 | 97.020 | 97.160 | 78 |
| 57.147 | 67.573 | 71.207 | 80.247 | 81.453 | 86.347 | 95.680 | 95.840 | 96.127 | 97.120 | 97.060 | 97.200 | 79 |
| 57.127 | 67.553 | 71.187 | 80.227 | 81.433 | 86.327 | 95.660 | 95.820 | 96.107 | 97.100 | 97.040 | 97.180 | 80 |

**Table 7:** Results from combining different variants of minimum sizes and fill percentage rectangle approach. The simulation uses 150 cluster in the detector and beam size 4mm, which results in the highest perfectly reconstructed cluster to be 97.580%. The best score is a combination of fill percentage 72% and minimum size 17. Each value is an average of 100 simulations.

| 0 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | Fill p. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 96.230 | 96.230 | 96.230 | 96.230 | 96.230 | 96.230 | 96.220 | 96.200 | 96.200 | 96.200 | 96.190 | 96.190 | 55 |
| 95.720 | 96.255 | 96.255 | 96.255 | 96.255 | 96.245 | 96.235 | 96.215 | 96.215 | 96.215 | 96.205 | 96.205 | 56 |
| 95.695 | 96.230 | 96.230 | 96.320 | 96.320 | 96.310 | 96.300 | 96.280 | 96.260 | 96.260 | 96.250 | 96.215 | 57 |
| 95.835 | 96.370 | 96.370 | 96.460 | 96.460 | 96.450 | 96.440 | 96.420 | 96.400 | 96.400 | 96.370 | 96.335 | 58 |
| 95.835 | 96.375 | 96.375 | 96.465 | 96.465 | 96.455 | 96.445 | 96.425 | 96.395 | 96.395 | 96.365 | 96.330 | 59 |
| 95.845 | 96.385 | 96.385 | 96.475 | 96.475 | 96.465 | 96.455 | 96.435 | 96.405 | 96.405 | 96.375 | 96.340 | 60 |
| 95.930 | 96.470 | 96.470 | 96.550 | 96.550 | 96.540 | 96.590 | 96.570 | 96.540 | 96.520 | 96.490 | 96.455 | 61 |
| 95.940 | 96.480 | 96.480 | 96.560 | 96.560 | 96.550 | 96.600 | 96.580 | 96.550 | 96.530 | 96.500 | 96.465 | 62 |
| 94.780 | 95.320 | 95.320 | 95.400 | 96.605 | 96.595 | 96.645 | 96.625 | 96.595 | 96.575 | 96.545 | 96.510 | 63 |
| 94.855 | 95.395 | 95.395 | 95.475 | 96.680 | 96.670 | 96.720 | 96.700 | 96.670 | 96.650 | 96.620 | 96.585 | 64 |
| 94.925 | 95.465 | 95.465 | 95.545 | 96.750 | 96.740 | 96.790 | 96.770 | 96.740 | 96.720 | 96.660 | 96.625 | 65 |
| 94.840 | 95.380 | 95.380 | 95.460 | 96.665 | 96.655 | 96.705 | 96.835 | 96.805 | 96.785 | 96.725 | 96.690 | 66 |
| 89.765 | 91.915 | 95.460 | 95.540 | 96.745 | 96.735 | 96.785 | 96.915 | 96.885 | 96.865 | 96.730 | 96.695 | 67 |
| 89.780 | 91.930 | 95.475 | 95.555 | 96.760 | 96.750 | 96.800 | **96.930** | 96.900 | 96.880 | 96.745 | 96.710 | 68 |
| 84.675 | 86.825 | 90.370 | 90.450 | 91.655 | 96.655 | 96.705 | 96.835 | 96.805 | 96.785 | 96.650 | 96.790 | 69 |
| 84.690 | 86.840 | 90.385 | 90.465 | 91.670 | 96.670 | 96.720 | 96.850 | 96.820 | 96.800 | 96.665 | 96.805 | 70 |
| 84.535 | 86.685 | 90.230 | 90.310 | 91.515 | 96.515 | 96.565 | 96.695 | 96.900 | 96.880 | 96.745 | 96.805 | 71 |
| 84.565 | 86.715 | 90.260 | 90.340 | 91.545 | 96.545 | 96.595 | 96.725 | **96.930** | 96.910 | 96.775 | 96.835 | 72 |
| 84.315 | 86.465 | 90.010 | 90.090 | 91.295 | 96.295 | 96.345 | 96.495 | 96.700 | 96.680 | 96.545 | 96.605 | 73 |
| 84.250 | 86.400 | 89.945 | 90.025 | 91.230 | 96.260 | 96.310 | 96.460 | 96.665 | 96.645 | 96.510 | 96.570 | 74 |
| 84.260 | 86.410 | 89.955 | 90.035 | 91.240 | 96.270 | 96.320 | 96.470 | 96.675 | 96.655 | 96.520 | 96.580 | 75 |
| 64.435 | 67.395 | 70.940 | 80.095 | 81.300 | 86.330 | 95.440 | 95.590 | 95.795 | 96.725 | 96.590 | 96.650 | 76 |
| 64.300 | 67.260 | 70.805 | 79.960 | 81.165 | 86.195 | 95.305 | 95.455 | 95.660 | 96.590 | 96.445 | 96.505 | 77 |
| 57.260 | 67.270 | 70.815 | 79.970 | 81.175 | 86.205 | 95.315 | 95.465 | 95.660 | 96.590 | 96.445 | 96.505 | 78 |
| 57.320 | 67.330 | 70.875 | 80.030 | 81.235 | 86.265 | 95.375 | 95.525 | 95.720 | 96.650 | 96.505 | 96.565 | 79 |
| 57.325 | 67.335 | 70.880 | 80.035 | 81.240 | 86.270 | 95.380 | 95.530 | 95.725 | 96.655 | 96.510 | 96.570 | 80 |

**Table 8:** Results from combining different variants of minimum sizes and fill percentage rectangle approach. The simulation uses 200 cluster in the detector and beam size 4mm, which results in the highest perfectly reconstructed cluster to be 96.930%. There are 2 combination which results in the highest score, including fill percentage 72% and minimum size 14. Each value is an average of 100 simulations.

| 0 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | Fill p. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 95.388 | 95.388 | 95.388 | 95.388 | 95.388 | 95.388 | 95.380 | 95.372 | 95.372 | 95.372 | 95.356 | 95.356 | 55 |
| 94.900 | 95.420 | 95.420 | 95.420 | 95.420 | 95.400 | 95.392 | 95.384 | 95.384 | 95.384 | 95.368 | 95.368 | 56 |
| 94.936 | 95.456 | 95.456 | 95.532 | 95.532 | 95.512 | 95.504 | 95.496 | 95.484 | 95.484 | 95.468 | 95.392 | 57 |
| 95.104 | 95.624 | 95.624 | 95.700 | 95.700 | 95.680 | 95.672 | 95.664 | 95.652 | 95.652 | 95.620 | 95.544 | 58 |
| 95.128 | 95.656 | 95.656 | 95.732 | 95.732 | 95.712 | 95.704 | 95.696 | 95.660 | 95.660 | 95.628 | 95.552 | 59 |
| 95.144 | 95.672 | 95.672 | 95.748 | 95.748 | 95.728 | 95.720 | 95.712 | 95.676 | 95.676 | 95.644 | 95.568 | 60 |
| 95.224 | 95.752 | 95.752 | 95.820 | 95.820 | 95.800 | 95.860 | 95.852 | 95.816 | 95.792 | 95.760 | 95.676 | 61 |
| 95.252 | 95.780 | 95.780 | 95.848 | 95.848 | 95.828 | 95.888 | 95.880 | 95.844 | 95.820 | 95.788 | 95.704 | 62 |
| 94.148 | 94.676 | 94.676 | 94.744 | 95.940 | 95.920 | 95.980 | 95.972 | 95.936 | 95.912 | 95.880 | 95.796 | 63 |
| 94.252 | 94.780 | 94.780 | 94.848 | 96.044 | 96.024 | 96.084 | 96.076 | 96.040 | 96.016 | 95.984 | 95.900 | 64 |
| 94.372 | 94.900 | 94.900 | 94.968 | 96.164 | 96.144 | 96.204 | 96.196 | 96.160 | 96.136 | 96.064 | 95.980 | 65 |
| 94.324 | 94.852 | 94.852 | 94.920 | 96.116 | 96.096 | 96.156 | 96.292 | 96.256 | 96.232 | 96.160 | 96.076 | 66 |
| 89.300 | 91.428 | 94.972 | 95.040 | 96.236 | 96.216 | 96.276 | 96.412 | 96.376 | 96.352 | 96.176 | 96.092 | 67 |
| 89.336 | 91.464 | 95.008 | 95.076 | 96.272 | 96.252 | 96.312 | 96.448 | 96.412 | 96.388 | 96.212 | 96.128 | 68 |
| 84.420 | 86.548 | 90.092 | 90.160 | 91.356 | 96.232 | 96.292 | 96.428 | 96.392 | 96.368 | 96.192 | 96.208 | 69 |
| 84.448 | 86.576 | 90.120 | 90.188 | 91.384 | 96.260 | 96.320 | 96.456 | 96.420 | 96.396 | 96.220 | 96.236 | 70 |
| 84.320 | 86.448 | 89.992 | 90.060 | 91.256 | 96.132 | 96.192 | 96.328 | 96.512 | 96.488 | 96.312 | 96.252 | 71 |
| 84.380 | 86.508 | 90.052 | 90.120 | 91.316 | 96.192 | 96.252 | 96.388 | **96.572** | 96.548 | 96.372 | 96.312 | 72 |
| 84.164 | 86.292 | 89.836 | 89.904 | 91.100 | 95.976 | 96.036 | 96.188 | 96.372 | 96.348 | 96.172 | 96.112 | 73 |
| 84.112 | 86.240 | 89.784 | 89.852 | 91.048 | 95.952 | 96.012 | 96.164 | 96.348 | 96.324 | 96.148 | 96.088 | 74 |
| 84.120 | 86.248 | 89.792 | 89.860 | 91.056 | 95.960 | 96.020 | 96.172 | 96.356 | 96.332 | 96.156 | 96.096 | 75 |
| 64.504 | 67.428 | 70.972 | 80.080 | 81.276 | 86.180 | 95.220 | 95.372 | 95.556 | 96.432 | 96.256 | 96.196 | 76 |
| 64.384 | 67.308 | 70.852 | 79.960 | 81.156 | 86.060 | 95.100 | 95.252 | 95.436 | 96.312 | 96.128 | 96.068 | 77 |
| 57.472 | 67.332 | 70.876 | 79.984 | 81.180 | 86.084 | 95.124 | 95.276 | 95.436 | 96.312 | 96.128 | 96.068 | 78 |
| 57.544 | 67.404 | 70.948 | 80.056 | 81.252 | 86.156 | 95.196 | 95.348 | 95.508 | 96.384 | 96.200 | 96.140 | 79 |
| 57.564 | 67.424 | 70.968 | 80.076 | 81.272 | 86.176 | 95.216 | 95.368 | 95.528 | 96.404 | 96.220 | 96.160 | 80 |

**Table 9:** Results from combining different variants of minimum sizes and fill percentage rectangle approach. The simulation uses 250 cluster in the detector and beam size 4mm, which results in the highest perfectly reconstructed cluster to be 96.572%. The best combination was fill percentage 72% and minimum size 14. Each value is an average of 100 simulations.

| 0 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | Fill p. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 94.420 | 94.420 | 94.420 | 94.420 | 94.420 | 94.420 | 94.413 | 94.413 | 94.413 | 94.413 | 94.393 | 94.393 | 55 |
| 93.983 | 94.460 | 94.460 | 94.460 | 94.460 | 94.443 | 94.437 | 94.437 | 94.437 | 94.437 | 94.417 | 94.417 | 56 |
| 93.983 | 94.460 | 94.460 | 94.557 | 94.557 | 94.540 | 94.533 | 94.533 | 94.523 | 94.523 | 94.503 | 94.437 | 57 |
| 94.137 | 94.613 | 94.613 | 94.710 | 94.710 | 94.693 | 94.687 | 94.687 | 94.677 | 94.677 | 94.637 | 94.570 | 58 |
| 94.157 | 94.643 | 94.643 | 94.740 | 94.740 | 94.723 | 94.717 | 94.717 | 94.673 | 94.673 | 94.633 | 94.567 | 59 |
| 94.170 | 94.657 | 94.657 | 94.753 | 94.753 | 94.737 | 94.730 | 94.730 | 94.687 | 94.687 | 94.647 | 94.580 | 60 |
| 94.327 | 94.813 | 94.813 | 94.897 | 94.897 | 94.880 | 94.957 | 94.957 | 94.913 | 94.880 | 94.840 | 94.753 | 61 |
| 94.357 | 94.843 | 94.843 | 94.927 | 94.927 | 94.910 | 94.987 | 94.987 | 94.943 | 94.910 | 94.870 | 94.783 | 62 |
| 93.293 | 93.780 | 93.780 | 93.863 | 95.050 | 95.033 | 95.110 | 95.110 | 95.067 | 95.020 | 94.980 | 94.893 | 63 |
| 93.407 | 93.893 | 93.893 | 93.977 | 95.163 | 95.147 | 95.223 | 95.223 | 95.180 | 95.133 | 95.093 | 95.007 | 64 |
| 93.557 | 94.043 | 94.043 | 94.127 | 95.313 | 95.297 | 95.373 | 95.373 | 95.330 | 95.283 | 95.200 | 95.113 | 65 |
| 93.543 | 94.030 | 94.030 | 94.113 | 95.300 | 95.283 | 95.360 | 95.480 | 95.437 | 95.390 | 95.307 | 95.220 | 66 |
| 88.597 | 90.690 | 94.233 | 94.317 | 95.503 | 95.487 | 95.563 | 95.683 | 95.640 | 95.593 | 95.360 | 95.273 | 67 |
| 88.640 | 90.733 | 94.277 | 94.360 | 95.547 | 95.530 | 95.607 | 95.727 | 95.683 | 95.637 | 95.403 | 95.317 | 68 |
| 83.843 | 85.937 | 89.480 | 89.563 | 90.750 | 95.520 | 95.597 | 95.717 | 95.673 | 95.627 | 95.393 | 95.403 | 69 |
| 83.867 | 85.960 | 89.503 | 89.587 | 90.773 | 95.543 | 95.620 | 95.740 | 95.697 | 95.650 | 95.417 | 95.427 | 70 |
| 83.860 | 85.953 | 89.497 | 89.580 | 90.767 | 95.537 | 95.613 | 95.733 | 95.863 | 95.817 | 95.583 | 95.467 | 71 |
| 83.963 | 86.057 | 89.600 | 89.683 | 90.870 | 95.640 | 95.717 | 95.837 | **95.967** | 95.907 | 95.673 | 95.557 | 72 |
| 83.780 | 85.873 | 89.417 | 89.500 | 90.687 | 95.457 | 95.533 | 95.670 | 95.800 | 95.740 | 95.507 | 95.390 | 73 |
| 83.747 | 85.840 | 89.383 | 89.467 | 90.653 | 95.440 | 95.517 | 95.653 | 95.783 | 95.723 | 95.490 | 95.373 | 74 |
| 83.767 | 85.860 | 89.403 | 89.487 | 90.673 | 95.460 | 95.537 | 95.673 | 95.803 | 95.743 | 95.510 | 95.393 | 75 |
| 64.510 | 67.403 | 70.947 | 80.020 | 81.207 | 85.993 | 94.830 | 94.967 | 95.097 | 95.917 | 95.683 | 95.567 | 76 |
| 64.403 | 67.297 | 70.840 | 79.913 | 81.100 | 85.887 | 94.723 | 94.860 | 94.990 | 95.810 | 95.570 | 95.453 | 77 |
| 57.557 | 67.317 | 70.860 | 79.933 | 81.120 | 85.907 | 94.743 | 94.880 | 94.990 | 95.810 | 95.570 | 95.453 | 78 |
| 57.643 | 67.403 | 70.947 | 80.020 | 81.207 | 85.993 | 94.830 | 94.967 | 95.077 | 95.897 | 95.657 | 95.540 | 79 |
| 57.677 | 67.437 | 70.980 | 80.053 | 81.240 | 86.027 | 94.863 | 95.000 | 95.110 | 95.930 | 95.690 | 95.573 | 80 |

**Table 10:** Results from combining different variants of minimum sizes and fill percentage rectangle approach. The simulation uses 300 cluster in the detector and beam size 4mm, which results in the highest perfectly reconstructed cluster to be 95.967%. The best combination was fill percentage 72% and minimum size 14. Each value is an average of 100 simulations.

# Appendix C

| 0 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | Clusters |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 100.00 | 100.00 | 100.00 | 100.00 | 100.00 | 100.00 | 100.00 | 100.00 | 100.00 | 100.00 | 100.00 | 100.00 | 0 |
| 94.360 | 94.800 | 98.840 | 98.920 | 99.080 | 99.280 | 99.320 | 99.480 | 99.560 | 99.680 | 99.520 | 99.560 | 25 |
| 93.820 | 94.380 | 98.200 | 98.260 | 98.560 | 98.740 | 98.780 | 98.900 | 98.980 | 99.000 | 98.900 | 98.880 | 50 |
| 93.880 | 94.427 | 98.160 | 98.213 | 98.520 | 98.667 | 98.680 | 98.760 | 98.800 | 98.867 | 98.747 | 98.733 | 75 |
| 93.310 | 93.890 | 97.630 | 97.720 | 98.050 | 98.210 | 98.270 | 98.340 | 98.360 | 98.410 | 98.320 | 98.280 | 100 |
| 93.200 | 93.720 | 97.344 | 97.440 | 97.760 | 97.880 | 97.928 | 98.000 | 97.992 | 98.016 | 97.912 | 97.872 | 125 |
| 92.933 | 93.453 | 97.073 | 97.160 | 97.453 | 97.580 | 97.627 | 97.673 | 97.680 | 97.720 | 97.620 | 97.587 | 150 |
| 92.469 | 92.989 | 96.600 | 96.674 | 96.977 | 97.097 | 97.166 | 97.206 | 97.217 | 97.246 | 97.149 | 97.086 | 175 |
| 92.265 | 92.800 | 96.325 | 96.415 | 96.695 | 96.815 | 96.875 | 96.925 | 96.920 | 96.935 | 96.820 | 96.730 | 200 |
| 92.098 | 92.627 | 96.124 | 96.209 | 96.480 | 96.596 | 96.653 | 96.711 | 96.702 | 96.720 | 96.582 | 96.493 | 225 |
| 91.868 | 92.392 | 95.912 | 95.996 | 96.248 | 96.356 | 96.416 | 96.476 | 96.456 | 96.464 | 96.324 | 96.188 | 250 |
| 91.596 | 92.102 | 95.615 | 95.705 | 95.953 | 96.051 | 96.127 | 96.189 | 96.178 | 96.164 | 96.007 | 95.851 | 275 |
| 91.327 | 91.807 | 95.327 | 95.423 | 95.660 | 95.747 | 95.823 | 95.880 | 95.840 | 95.783 | 95.597 | 95.437 | 300 |
| 91.089 | 91.588 | 95.102 | 95.203 | 95.431 | 95.498 | 95.569 | 95.622 | 95.566 | 95.480 | 95.271 | 95.108 | 325 |
| 90.746 | 91.231 | 94.751 | 94.854 | 95.089 | 95.154 | 95.223 | 95.271 | 95.197 | 95.106 | 94.894 | 94.717 | 350 |
| 90.488 | 90.987 | 94.488 | 94.595 | 94.821 | 94.891 | 94.955 | 94.995 | 94.920 | 94.797 | 94.544 | 94.336 | 375 |
| 90.185 | 90.688 | 94.190 | 94.305 | 94.522 | 94.590 | 94.645 | 94.668 | 94.603 | 94.465 | 94.207 | 93.978 | 400 |
| 89.878 | 90.369 | 93.852 | 93.969 | 94.191 | 94.256 | 94.304 | 94.325 | 94.249 | 94.115 | 93.826 | 93.581 | 425 |
| 89.613 | 90.102 | 93.576 | 93.691 | 93.922 | 93.991 | 94.027 | 94.027 | 93.960 | 93.818 | 93.536 | 93.273 | 450 |
| 89.320 | 89.813 | 93.242 | 93.352 | 93.568 | 93.625 | 93.655 | 93.646 | 93.564 | 93.396 | 93.091 | 92.811 | 475 |
| 89.146 | 89.652 | 93.018 | 93.132 | 93.348 | 93.398 | 93.434 | 93.422 | 93.348 | 93.172 | 92.854 | 92.564 | 500 |

**Table 11:** Results from simulation of the asymmetric cut using different minimum sizes. Each value is an average of 100 simulations.

# Appendix D

| 0 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | Clusters |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 100.00 | 100.00 | 100.00 | 100.00 | 100.00 | 100.00 | 100.00 | 100.00 | 100.00 | 100.00 | 100.00 | 100.000 | 0 |
| 99.680 | 99.680 | 99.680 | 99.680 | 99.680 | 99.680 | 99.680 | 99.720 | 99.720 | 99.760 | 99.680 | 99.680 | 25 |
| 98.920 | 98.920 | 98.920 | 98.920 | 98.920 | 98.920 | 98.920 | 98.960 | 98.920 | 98.940 | 98.880 | 98.840 | 50 |
| 98.853 | 98.853 | 98.853 | 98.853 | 98.853 | 98.853 | 98.853 | 98.880 | 98.853 | 98.867 | 98.827 | 98.800 | 75 |
| 98.420 | 98.420 | 98.420 | 98.420 | 98.420 | 98.420 | 98.420 | 98.450 | 98.410 | 98.410 | 98.370 | 98.330 | 100 |
| 97.912 | 97.912 | 97.912 | 97.912 | 97.912 | 97.912 | 97.912 | 97.936 | 97.904 | 97.904 | 97.856 | 97.808 | 125 |
| 97.620 | 97.620 | 97.620 | 97.620 | 97.620 | 97.620 | 97.627 | 97.647 | 97.607 | 97.620 | 97.580 | 97.540 | 150 |
| 97.211 | 97.211 | 97.211 | 97.211 | 97.211 | 97.211 | 97.217 | 97.234 | 97.177 | 97.166 | 97.131 | 97.051 | 175 |
| 96.905 | 96.905 | 96.905 | 96.905 | 96.905 | 96.905 | 96.910 | 96.935 | 96.865 | 96.855 | 96.795 | 96.695 | 200 |
| 96.649 | 96.653 | 96.653 | 96.653 | 96.653 | 96.644 | 96.649 | 96.662 | 96.591 | 96.582 | 96.529 | 96.431 | 225 |
| 96.364 | 96.368 | 96.368 | 96.368 | 96.368 | 96.360 | 96.364 | 96.376 | 96.296 | 96.272 | 96.216 | 96.096 | 250 |
| 96.015 | 96.018 | 96.018 | 96.018 | 96.018 | 96.011 | 96.015 | 96.029 | 95.956 | 95.920 | 95.847 | 95.687 | 275 |
| 95.617 | 95.620 | 95.620 | 95.620 | 95.620 | 95.607 | 95.610 | 95.617 | 95.550 | 95.503 | 95.427 | 95.267 | 300 |
| 95.277 | 95.283 | 95.283 | 95.283 | 95.277 | 95.265 | 95.268 | 95.268 | 95.212 | 95.151 | 95.068 | 94.889 | 325 |
| 94.843 | 94.849 | 94.849 | 94.849 | 94.843 | 94.831 | 94.834 | 94.840 | 94.783 | 94.709 | 94.614 | 94.420 | 350 |
| 94.424 | 94.429 | 94.429 | 94.429 | 94.419 | 94.403 | 94.405 | 94.405 | 94.352 | 94.267 | 94.168 | 93.971 | 375 |
| 94.080 | 94.088 | 94.088 | 94.088 | 94.078 | 94.062 | 94.055 | 94.050 | 94.000 | 93.910 | 93.793 | 93.582 | 400 |
| 93.696 | 93.704 | 93.704 | 93.704 | 93.694 | 93.680 | 93.668 | 93.664 | 93.607 | 93.518 | 93.393 | 93.195 | 425 |
| 93.404 | 93.411 | 93.411 | 93.411 | 93.402 | 93.389 | 93.369 | 93.364 | 93.311 | 93.222 | 93.104 | 92.891 | 450 |
| 92.958 | 92.971 | 92.971 | 92.971 | 92.958 | 92.941 | 92.922 | 92.918 | 92.867 | 92.766 | 92.642 | 92.415 | 475 |
| 92.622 | 92.634 | 92.634 | 92.634 | 92.622 | 92.606 | 92.588 | 92.584 | 92.528 | 92.432 | 92.298 | 92.078 | 500 |

**Table 12:** Results from simulation of the inactive pixels cut using different minimum sizes. Each value is an average of 100 simulations.

# Appendix E

| A | I | F | AF | IF | AI | AIF | B | Clusters |
|---|---|---|---|---|---|---|---|---|
| 100.000 | 100.000 | 100.000 | 100.000 | 100.000 | 100.000 | 100.000 | 100.000 | 0 |
| 99.400 | 99.620 | 99.180 | 99.020 | 99.260 | 99.400 | 99.020 | 99.520 | 25 |
| 98.840 | 99.000 | 98.750 | 98.610 | 98.810 | 98.840 | 98.610 | 98.740 | 50 |
| 98.573 | 98.760 | 98.680 | 98.547 | 98.733 | 98.573 | 98.547 | 98.387 | 75 |
| 98.345 | 98.485 | 98.475 | 98.335 | 98.505 | 98.345 | 98.335 | 98.000 | 100 |
| 97.996 | 98.008 | 98.128 | 98.032 | 98.156 | 97.996 | 98.032 | 97.512 | 125 |
| 97.647 | 97.680 | 97.873 | 97.773 | 97.907 | 97.647 | 97.773 | 97.090 | 150 |
| 97.323 | 97.320 | 97.657 | 97.551 | 97.737 | 97.323 | 97.551 | 96.603 | 175 |
| 97.010 | 96.938 | 97.380 | 97.332 | 97.465 | 97.010 | 97.332 | 96.055 | 200 |
| 96.764 | 96.651 | 97.167 | 97.149 | 97.242 | 96.764 | 97.149 | 95.647 | 225 |
| 96.444 | 96.274 | 96.932 | 96.950 | 97.036 | 96.444 | 96.950 | 95.088 | 250 |
| 96.147 | 95.925 | 96.698 | 96.758 | 96.807 | 96.147 | 96.758 | 94.605 | 275 |
| 95.818 | 95.538 | 96.398 | 96.522 | 96.528 | 95.818 | 96.522 | 94.092 | 300 |
| 95.529 | 95.194 | 96.132 | 96.298 | 96.268 | 95.529 | 96.298 | 93.615 | 325 |
| 95.263 | 94.854 | 95.900 | 96.096 | 96.043 | 95.263 | 96.096 | 93.163 | 350 |
| 94.976 | 94.468 | 95.645 | 95.853 | 95.787 | 94.976 | 95.853 | 92.672 | 375 |
| 94.619 | 94.093 | 95.338 | 95.564 | 95.490 | 94.619 | 95.564 | 92.188 | 400 |
| 94.247 | 93.673 | 95.021 | 95.296 | 95.188 | 94.247 | 95.296 | 91.632 | 425 |
| 93.931 | 93.323 | 94.736 | 95.022 | 94.907 | 93.931 | 95.022 | 91.211 | 450 |
| 93.582 | 92.933 | 94.434 | 94.753 | 94.619 | 93.582 | 94.753 | 90.689 | 475 |
| 93.331 | 92.576 | 94.190 | 94.548 | 94.376 | 93.331 | 94.548 | 90.272 | 500 |

**Table 13:** Results from simulation with mixtures of all cuts. Each value is an average of 200 simulations. The values have the following abbreviation: A=Asymmetric, I=Inactive pixels, F=fill percentage square approach, and B=basic method.