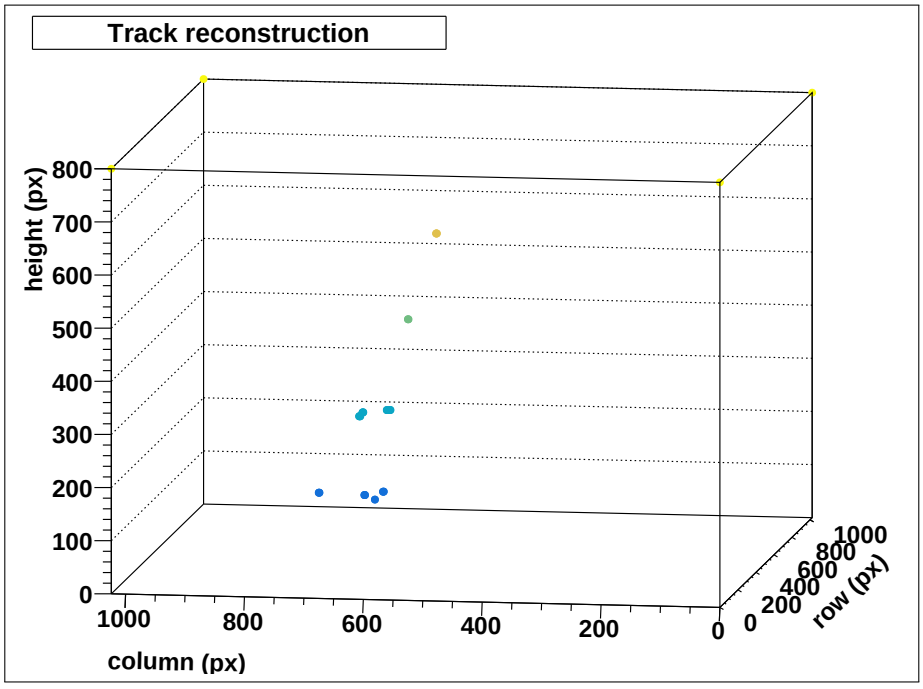
Universiteit Utrecht

Opleiding Natuur- en Sterrenkunde

Tests of a new digital calorimeter prototype based on the ALPIDE pixel chip

Bachelor Thesis

Sebastiaan van Rijk



Supervisors:

Prof .Dr. Thomas Peitzmann
Subatomic Physics

Dr. ing. Naomi van der Kolk
Subatomic Physics

15 january 2020

Abstract

In the moments after the Big Bang the universe was in a state which we call a Quark Gluon Plasma. The study of this state of matter is the primary objective at the ALICE detector in the LHC. ALICE has a proposed upgrade called FoCal (Forward Calorimeter) which will give us a new window into this state of matter.
For the research and development of FoCal we built a prototype which uses the ALPIDE chip as the sensors. The ALPIDE chip can be configured using analog settings, which will influence the sensitivity of the chip. The influence of these settings on the noise rate and cluster size of cosmic ray particles has been measured.
From the cosmic ray particles that were detected there has been made a selection on the muons, which produce straight tracks in the prototype, to produce an alignment procedure. From the resulting alignment graphs and research it was found that the procedure used only gives a first order correction and a better alignment algorithm should be used.

Front page: This is a 3D event display of an electromagnetic shower in the prototype calorimeter.

Contents

1 Introduction

One of the most important unanswered questions we physicist have is: "What is matter and how did it come about?". The field which has the most promise of answering this question is the field of experimental particle physics. This is because in experimental particle physics we break matter into its most fundamental constituents and reproduce conditions of the early universe, slightly after the Big Bang.

To reproduce these conditions we use particle accelerators, which come in two varieties; circular and linear. Currently the particle accelerator which achieves the highest energy is the LHC (**L**arge **H**adron **C**ollider), which is a circular accelerator with a circumference of 27 km.

In the LHC either protons or lead nuclei get accelerated in opposite directions, to get them to collide in detectors. One such detector is ALICE (**A** **L**arge **I**on **C**ollider **E**xperiment), which was build to detect collisions between lead ions. It was predicted that these collisions would give rise to a new state of matter; Quark Gluon Plasma (QGP) which is theorized to be the state of the universe slightly after the Big Bang. The study of QGP is the primary research at ALICE.

Our current understanding of the QGP is limited and new measurements are needed to widen our understanding. FoCal (**Fo**rward **Cal**orimeter) is a proposed upgrade for ALICE. FoCal could provide such new measurements of the properties of the QGP created inside ALICE.

The research and development for FoCal is still ongoing and this thesis is part of testing a newly built prototype for FoCal. FoCal is partly made from silicon pixel chips. This is a new technology to be used in a calorimeter and a lot of preliminary testing and prototyping before the final installation in ALICE is needed.

My research on the pixel prototype focused on changing settings on the chips to see their effect on the noise rate. I also looked into using cosmic rays to determine the effect the settings had on the clustersize (which is correlated to the sensitivity) and using cosmic ray muons for alignment of the detector. I also participated in taking a week long data at the DESY test beam facility in Germany. The data from those measurements sadly will not be a part of this thesis.

2 Theoretical motivation

Atomic nuclei consist of protons and neutrons, which in term consist of three quarks; *uud* (proton) and *udd* (neutron). These quarks are kept together by the strong nuclear force which is mediated by gluons. The strong nuclear force is described by Quantum Chromodynamics (QCD). In QCD we assign color charge to quarks and gluons. QCD only allows configurations where the colors charges cancel each other out; i.e. red-green-blue and color-anticolor. This means that one can never observe a free gluon or quark, this is called confinement. A quark gluon plasma (QGP) is a state of matter in which quarks and gluons become deconfined and can move freely, analogous to how water has undergone a phase transition when it first was ice and became a fluid upon heating. To achieve this we need particle accelerators to collide heavy ions with enough energy to produce a QGP.
One way to study the properties of the QGP is through measurement of the momentum distribution of the quarks and gluons, the partons, in the moments after the collision. These distributions are called Parton Distribution Functions (PDF's) and are defined as the probability density of finding a parton with a longitudinal momentum fraction x at a resolution scale Q^2. These cannot be computed within a QCD framework and can only be determined experimentally.
Since we cannot measure the PDF's directly the best thing we can measure are photons emitted by the quarks and gluons. These direct photons can probe the initial state (the PDF's) of the QGP [1] [2]. To measure gluon saturation, which can happen in a QGP, we need to measure these thermal photons at forward rapidities[1]. This is why FoCal should be placed in the forward position, close to the beam pipe, where it will collect direct thermal photons from the QGP. A noise background from pion decay ($\pi^0 \to \gamma\gamma$) will also be present this is why we need a device that measures photon energy with good spatial resolution to distinguish the two.
This can be achieved by a MAPS (Monolithic Active Pixel Sensor) calorimeter, which allows for far greater resolution than previous calorimeters due to having a high granularity. Which means that the pixel size is small enough to allow for having a small spatial resolution since the spatial resolution is on the scale of a pixel. Electromagnetic calorimeters measure photons (and electrons) and their energy in the following manner:

When a photon enters material it will give pair production ($\gamma \to e^- + e^+$), which can only occur near a nucleus due to conservation of momentum and therefor only occurs in matter. The produced electrons and positrons in turn scatter inside the material producing more photons due to bremsstrahlung which in turn again give pair production. This goes on until the photons no longer have enough energy for pair production and the electrons lose the rest of their energy through ionization. The entire process will produce an electromagnetic shower, see Fig. 1 for a schematic. In Fig. 1 the χ_0 is the radiation length, the length scale in which pair production occurs.
From the total deposited energy by the particle we can determine the energy of the initial photon. The total deposited energy of the particle is related to the total number of hits in the calorimeter, which is the number of pixels that are turned on in the case of a pixel calorimeter.

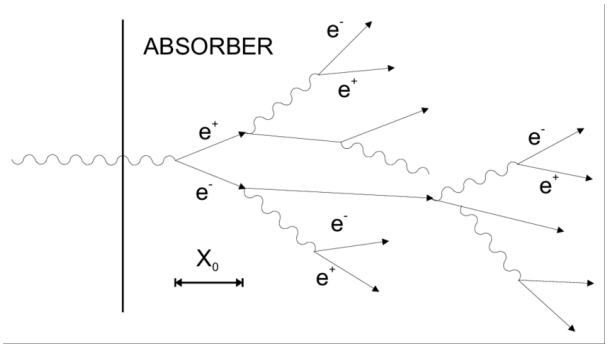


Figure 1: Schematic diagram of an electromagnetic shower.

3 Experimental setup

Our FoCal prototype detector, which we call the mTower (mini Tower), consists of an alternating sandwich structure of absorber layers (3 mm thick tungsten) and ALPIDE chips. The mTower is built in a modular fashion; each module consists of one tungsten layer with on top 2 ALPIDE chips. The final mTower prototype will have 24 modules, totalling 48 ALPIDE chips and thus contains over 25 million pixels. For the results in this thesis the mTower consisted of 4 layers (module numbers: $1, 8, 3$ and 9),see Fig. 2 label A, for the noise and in case of alignment of 7 layers.
The prototype is read out by two FPGA boards; the RUv0 and RUv2 boards, see Fig. 2 label C, which are made for the ITS (Inner Tracking System) in the ALICE detector. The trigger for measuring cosmics is made from two pieces of BC-420 plastic scintillator, see Fig. 2 label B,which produce a flash of light when ionizing radiation comes through. There is one on the top and one on the bottom both are read out by silicon photomultipliers and an incidence logic module. The reason why there is one on top and one on the bottom is to be sure that the trigger only occurs for cosmic ray particles that pass through the entire detector.

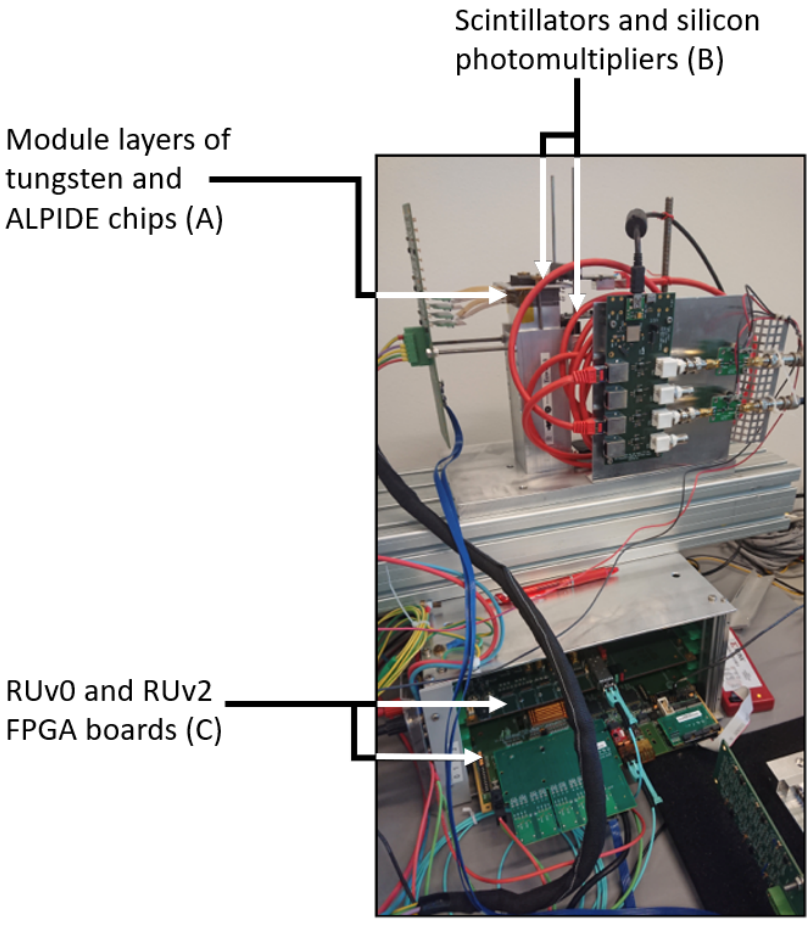


Figure 2: The mTower prototype of 4 layers.

For pedestal noise measurement experiments a random trigger is needed, for which we used a function generator.

3.1 ALPIDE chip

The ALPIDE chip is a MAPS (Monolithic Active Pixel Sensor) consisting of 512 by 1024 pixels which are roughly squares of 30 μm by 30 μm, giving the whole chip a size of about 3 cm by 1.5 cm. It is a MAPS in the sense that it works by having a negatively doped silicon layer, called the epitaxial layer, where charge is liberated when a charged particle moves through it. The liberated charges diffuse through the layer and end up at the diode where it activates the pixel if the amount of charge exceeds the threshold, see Fig. 3. Due to this diffusion more than one diode, and thus more than one pixel, can be activated by the same ionizing particle.

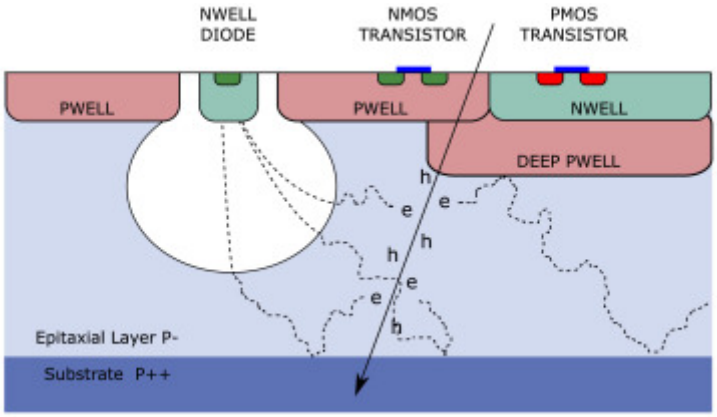


Figure 3: Cross section of a charged particle track through the ALPDIDE chip[3].

The threshold for how much charge is needed to activate the pixels in a chip can be controlled via the DAC settings of VCASN and ITHR, which can be seen in the circuitry of the pixels in Fig. 4. By changing VCASN and ITHR we can thus control the noise rate and sensitivity.

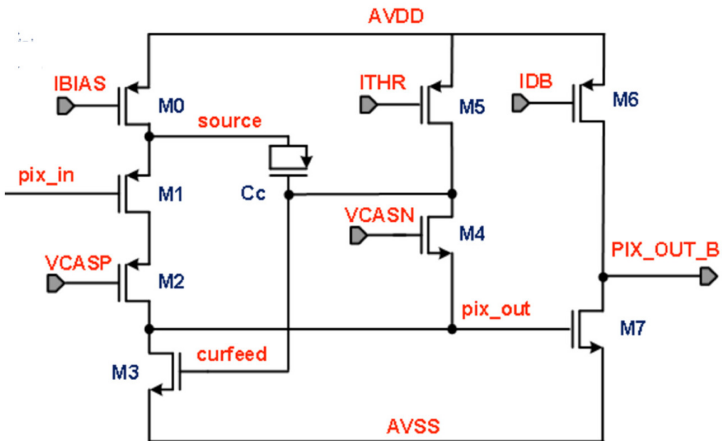


Figure 4: Part of the circuitry in a single pixel.

The noise occurs from thermal fluctuations in the pixels and from malfunctioning pixels which are one for a substantial amount of the time without, we call these hot pixels. These hot pixels are determined by doing a noise measurement and looking at which pixels are on for more than a specified percentage of the time, usually 10 % or 2 %.
Getting low enough noise rates is crucial in imaging electromagnetic showers with great detail. If the noise gets too high the signal from the shower is less clear and we get less well defined lateral and longitudinal shower profiles. Too high noise rate will also lower our spatial resolution due to earlier onset of detector saturation. The noise also influences the total number of hits in each event and thus influences the measurement of the energy of the incoming particle which is the primary purpose of the detector. From this we conclude that getting low enough noise rates is important.

4 Results

4.1 Noise

The noise rate of the ALPIDE chips was determined by doing multiple pedestal runs until enough statistics was gathered. From the data we could retrieve the hit distributions per setting. The settings used where ITHR constant at 51 and varying VCASN from 44 to 62 incrementally by 1. The following plots are the mean of these hit distributions, but normalized per pixel. So the mean number of hits per event divided by the number of chips and divided by the number of pixels of a single chip. This noise rate is thus the probability of a single pixel to be on during a single event. For the following noise results the mTower consisted of 8 ALPIDE chips (module numbers: $1, 8, 3$ and 9) and there were no pixel masks used, so a part of the noise is due to hot pixels. The error bars in the graphs are determined by dividing the standard deviation of the hit distribution by the square root of the number of events. The event rate was 100 Hz, so we took a measurement of the entire detector 100 times per second.

4.1.1 VCASN

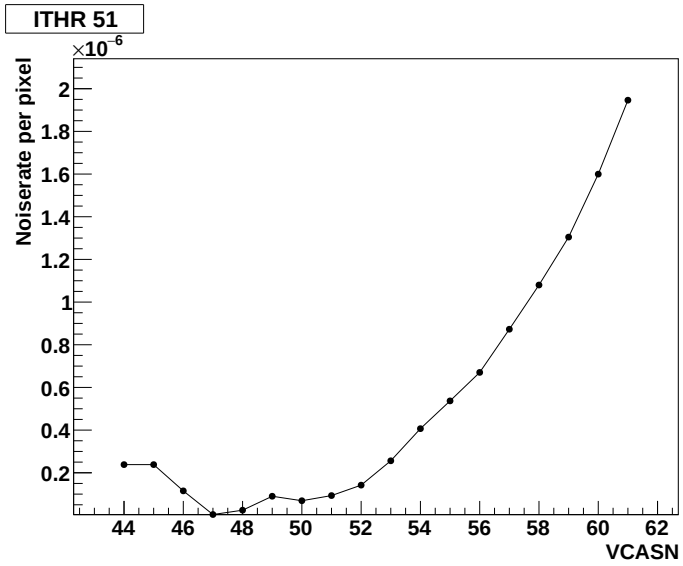


Figure 5: Noise rate per pixel at different VCASN setting of the ALPIDE chip. The ITHR setting was kept constant at 51.

We see in Fig. 5 that the noise rate grows exponentially with the VCASN DAC setting, from VCASN 44 to VCASN 60 the noise rate has increased by a about a factor of 10. On the low noise rate settings VCASN 44 to 50 we see some fluctuation, this is due to the noise rate being so low there.

4.1.2 ITHR

To get a complete picture of the noise rate for different settings we should also look into the effect of ITHR on the noise rate. We also did these ITHR varying runs for three different VCASN settings; VCASN 40, 50 and 60. The results are in Fig. 6, Fig. 7 and Fig. 8 respectively.

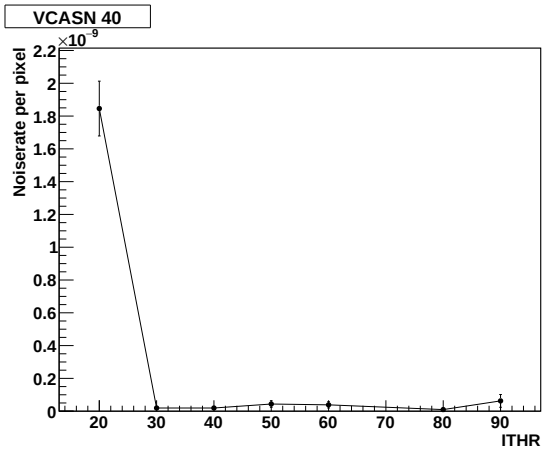


Figure 6: Noise rate per pixel at different ITHR settings, the VCASN setting kept constant at 40.

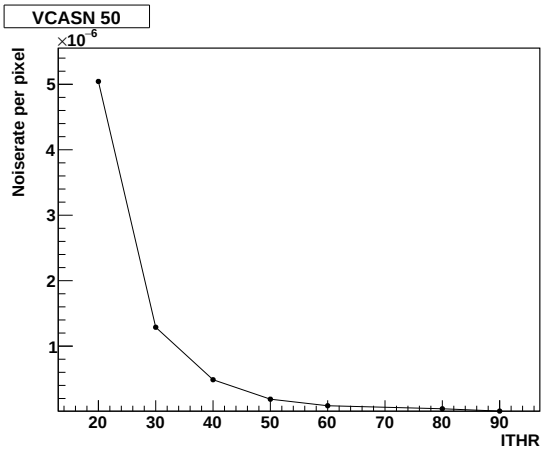


Figure 7: Noise rate per pixel at different ITHR settings, the VCASN setting kept constant at 50.

For the dataset where VCASN is kept constant at 60, Fig 8, I had to exclude 2 outliers, from two events, one for the ITHR at 20 and one for ITHR at 30. These were due to chip malfunctions and the hitmaps of the chip malfunctions are to be found in the appendix.

We see that noise rate is inversely proportional to the ITHR DAC setting; i.e. lower ITHR DAC setting values give higher noise rates. Again the noise rate increases exponantially but less than with the VCASN setting, here we see a increase of noise rate by about a factor of 5 in a span where VCASN would increase by a factor of around 10. We thus see that VCASN is more coarse way of tuning the noise rate and ITHR is a fine way of tuning.

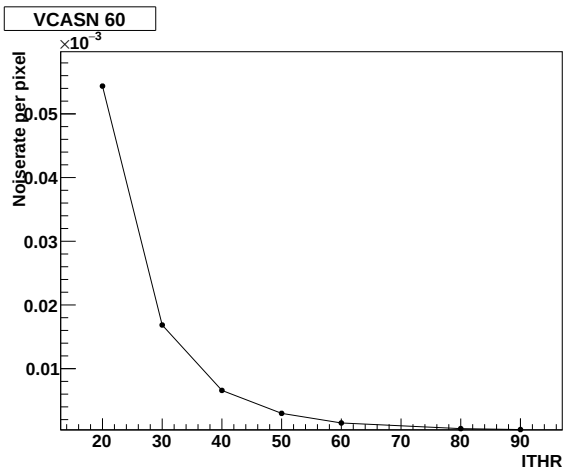


Figure 8: Noise rate per pixel at different ITHR settings, the VCASN setting kept constant at 60.

Now we have a complete picture of the noise rate phase space we can see which values of the DAC settings VCASN and ITHR we would like so we can minimize our noise rate while still having enough sensitivity to measure real particles coming through the detector.

At VCASN equal to 40, Fig. 6, the noise rate is really low ($\sim 10^{-9}$), but we can expect our sensitivity to be too low as well. At VCASN equal to 50 the noise rate is good ($\sim 10^{-6}$) and we don't sacrifice much sensitivity. At VCASN equal to 60 the noise rate is still good ($\sim 10^{-5}$). So a good value for VCASN is determined to be between 50 and 60. A good value for ITHR is determined to be between 40 and 60. The nominal settings recommended by the manual are VCASN 57 and ITHR 51. This will give a mean noise rate of $(8.73 \pm 0.01) * 10^{-7}$. Another combination of setting, recommended by a team from Bergen University, was VCASN 50 and ITHR 51. This results in a mean noise rate of $(6.90 \pm 0.04) * 10^{-8}$.
Fig. 9 show the hit distributions of the noise of the mentioned combinations of settings VCASN 50 ITHR 51 and VCASN 57 ITHR 51. The VCASN 57 ITHR 51 setting has a higher noise rate but there are never more than 10 noise pixels on during a single event with 8 ALPIDE chips, which is still acceptable as a noise rate.

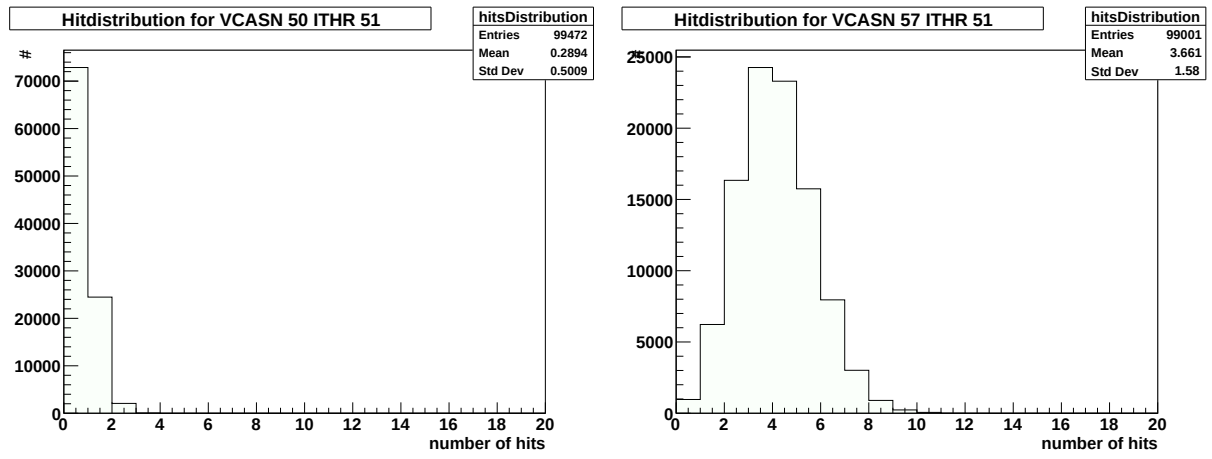


Figure 9: Hit distributions of noise measurements for VCASN 50 ITHR 51 (left) and VCASN 57 ITHR 51 (right) in 8 ALPIDE chips, y scales differ between graphs.

4.2 Cosmics

For determination of the sensitivity of the detector and the alignment we used cosmic rays. They are atomic nuclei [4] which collide with our atmosphere and decay into muons among other things. These muons are called cosmic ray muons, from here on they are what are meant by cosmics. The rate of cosmic muons at sea level is about 1 muon per square *cm* per minute. Using the fact that our detector has a surface area of 9 cm^2 we get to a rate of 9 muons through the top of the detector. But we only trigger for muons that pass both the top scintillator as well as the bottom scintillator, so we need to make an estimate based on our detector geometry. If we allow all paths that still go through both scintillators and assume the mTower to be a cylinder we arrive at the shape of a cone, see Fig. 10.

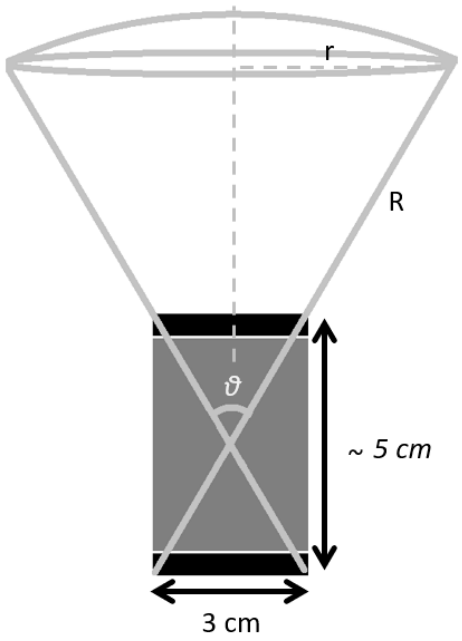


Figure 10: Schematic for the cone of the sky that gives muon paths that pass through both scintillators, the black squares. (not to scale)

From Fig. 10 we see that the angle θ will determine the area of the conic cap (where the cone intersects the halfdome) and in turn the ratio between the are of that cap and the area of half a dome of the same height will give an approximation to the fraction of muons that give a trigger. This is assuming that the distributions of incoming muons is isotropic in all directions and the detector is assumed to be a cylinder, which are both not true.
From trigonometry we arrive at a formula for θ.

$$\theta = 2 \times \arctan\left(\frac{0.5 \times width\ cm}{0.5 \times height\ cm}\right) \tag{1}$$

Eq. 1 gives us 61.93° for a height of 5 cm and width of 3 cm, we can approximate it to 60°. A formula for the muon trigger rate using the ratio of the area of a cone and a area of a half dome is given by Eq. 2.

$$\frac{d\mu}{dt}_{trig} = \frac{d\mu}{dt}_{sea} \times \frac{2\pi r^2\left(1 - cos\left(\frac{\theta}{2}\right)\right)}{2\pi R^2} \tag{2}$$

Where r is the radius of circle of the intersection of the cone and half dome, R is the radius of the half dome. We can relate r to R through Eq. 3.

$$r = R\sin\left(\frac{\theta}{2}\right) \tag{3}$$

We can now cancel terms to arrive at the final equation, Eq. 4.

$$\frac{d\mu}{dt}_{trig} = 9 \times \sin^2(30°)(1 - \cos(30°)) = 0.30\ muons\ per\ minute \tag{4}$$

Finally we approximate the rate to be 0.3 muons per minute or about a muon per 3 minutes. From this trigger rate we conclude that taking cosmic data for alignment takes a while to get enough statistics. So we collected cosmics over larger periods spanning days. During which we masked hot pixels, which we found by previous noise measurements and ignored all pixels which were on more than 10 % of the time.

4.2.1 Cluster size

In the following analysis we defined a cluster to be a group of pixels, meaning more than 1 pixel, that are connected by their neighbouring pixels.

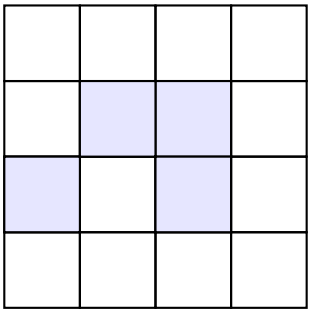


Figure 11: One cluster of size 4.

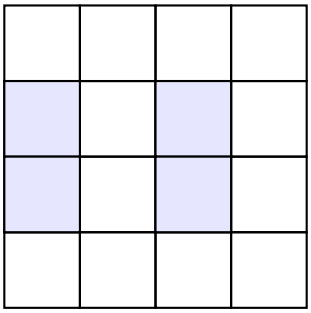


Figure 12: Two clusters of size 2.

The reason to look into the clustersize is that it is a measure of the sensitivity of the ALPIDE chips, since for a larger cluster more pixels need to be activated by the same amount of charge. It is also beneficial to have clusters formed by passing particles since this allows for better determination of the center of mass. But we do not want the clusters to be too large since than the clusters caused by different particles can start to merge when the number of particles that pass through the detector gets large, this we call saturation of the detector. The benefit from having clusters instead of isolated hits is that clusters allow for better determination of the center where the ionizing particle track is through the detector.

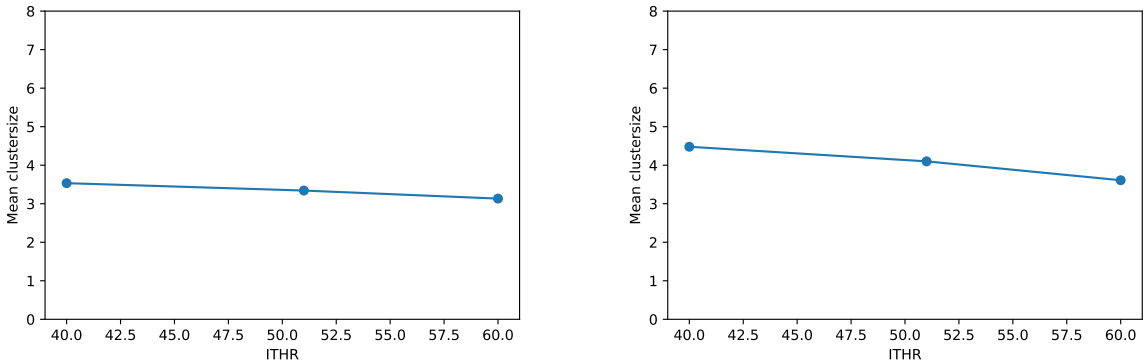


Figure 13: Mean cluster size due to cosmics for VCASN 50 (left) and VCASN 57 (right).

We see in Fig. 13 that for VCASN 57 clustersize is substantially bigger than VCASN 50, the ITHR setting changes the clustersize but less so. This confirms that VCASN is the coarse threshold setting and ITHR the fine setting. It could also be seen in the distributions, Fig 14 of the cluster size that a cluster size of an even number of pixels is more likely than clusters of odd number of pixels. The reason for this was assumed to be due to geometry of the pixels and the isotropic nature of the charge diffusion in through the pixel layer.
If the sensitivity went up due to lowering of ITHR the clusters went from primarily size 2 to primarily size 4.

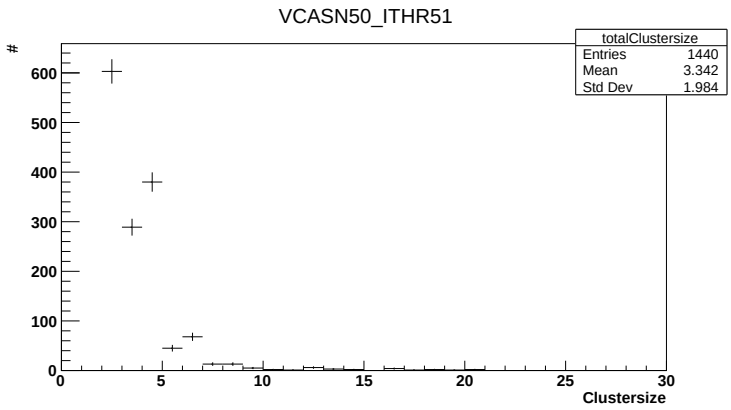


Figure 14: Distribution of cluster size of cosmics for setting VCASN 50 ITHR 51.

4.2.2 Alignment

For alignment we first need to use a classification algorithm since not every cosmic that passed through the detector is useful for alignment. We have three different classes; muons which produce a straight track through the detector, electrons which produces showers and events where multiple particles went through. For alignment we can only use events with a single straight track, the single muon events. So we only accept events where there is not more than one cluster in a single module, since otherwise there are multiple particles. And there must be at least a cluster in two different modules, so a straight line can be drawn.
From the fact that noise never gave a cluster but only single pixels we can immediately say that any cluster must have been due to a particle.

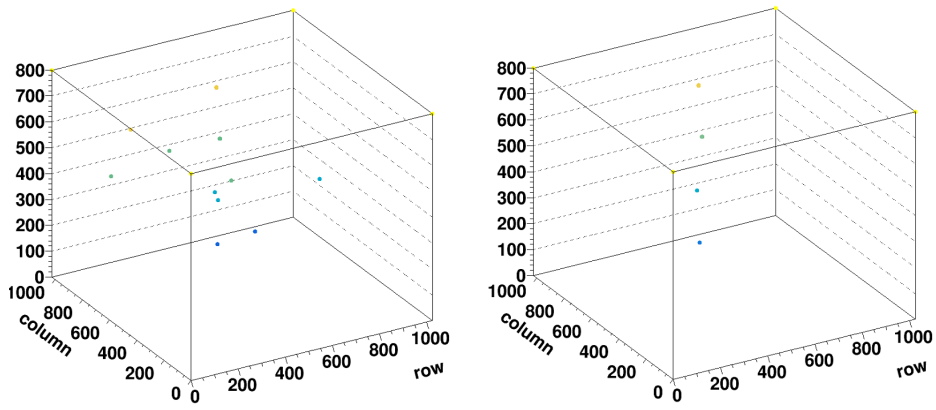


Figure 15: Event displays for all hits in a single event (left) and one where all the isolated hits are filtered and leave just the clusters (right), revealing a straight track from a muon. (axes are in pixels)

For fitting the straight lines we first need to know which chips are on top of one another. It was determined that for the prototype consisting of 7 layers the configuration of the lanes (chip numbers) was as follows:

Table 1: The internal orientation of the lane numbers for the mTower prototype consisting of 7 layers, 2 ALPIDE chips per layer.

6	7
4	5
0	3
2	1
13	12
11	10
9	8

We do not need to know which stack of chips was truly on which side in the physical prototype, only how they are oriented in respect to each other. Inside a module also one chip is rotated 180 degrees so you need to change coordinate systems from one side of the detector to get straight lines that cross from one side to the other side of the detector.
To get the best line fit we need to fit a line through the center of the clusters, the center is determined by using the center of mass formula.

$$(x, y) = \frac{\sum_{n=1}^{p}(x_n + \frac{1}{2}, y_n + \frac{1}{2})}{\sum_{n=1}^{p} n} \tag{5}$$

So I summed all the x coordinates that occur and divided by the number of pixels in a cluster to get the averaged x coordinate, the factor half is because we want the center of the pixel to count as the coordinate. I repeated this for the y coordinate to get the coordinates of the cluster center.

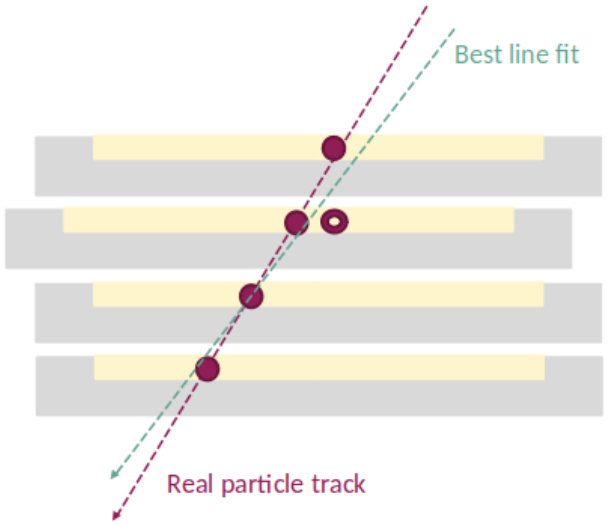


Figure 16: Schematic of a best line fit.

After fitting a straight line through the center of mass points I calculated the distance of the cluster in lane A between what coordinate the line crosses lane A and did that for all lanes where there is a cluster. This is as close we can get to the distance we want to measure which is in Fig. 16 the distance between the open red circle and the red dot, the red dot is where the cluster is in the chip but due to misalignment of the chip the coordinates are at the position of the open red circle.
These distances are called the residual, if you have enough statistics (a lot of straight muon tracks to fit) the residuals will start to form a Gaussian distribution. The mean of this Gaussian distribution is the offset and if we change the coordinates of lane A by minus the residual of lane A and redo the entire alignment algorithm the residual Gaussian should be centered at zero. The residuals were determined for a prototype consisting of 7 layers (14 chips):

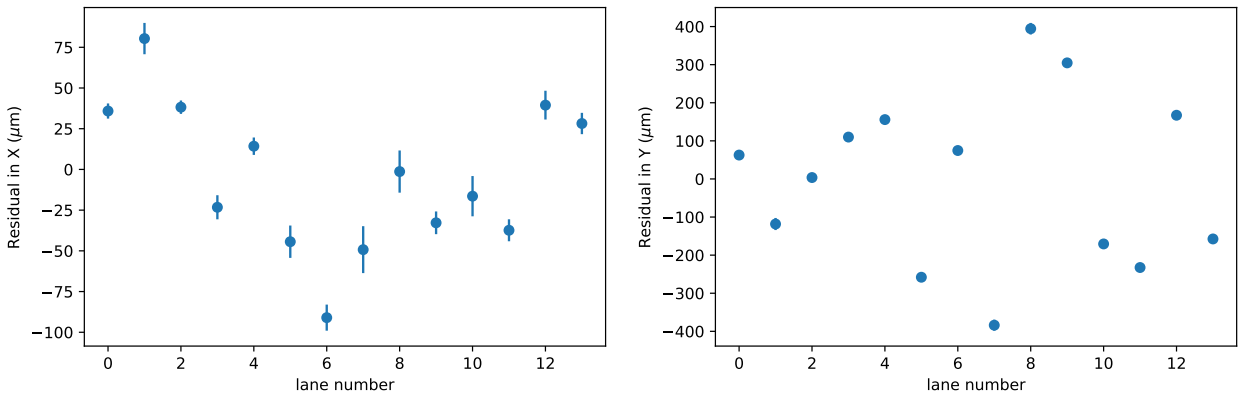


Figure 17: The residuals after fitting a large number of straight tracks,x translational direction on the left and y translational direction on the right. The scales of the graphs are not the same.

In Fig. 17 we see that is a difference between the residuals in the x and y coordinates. This difference becomes more apparent when we look at the residual plots for a single lane, lane 1 for example:

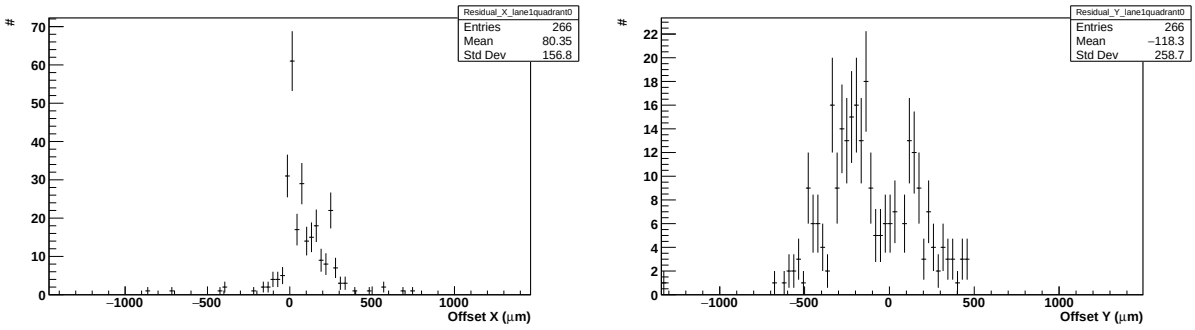


Figure 18: Residual plots for lane 1, x on the left and y on the right.

The y coordinate is on the short axis of the ALPIDE chip and the x axis is on the long axis of the ALPIDE chip. We would expect in an ideal scenario of just small internal translational offset among the chips for the graphs in Fig. 18 to be Gaussian distributions centered around a value that is not more than 150 μm (5 pixels) from 0. Clearly this not the case for the right plot in Fig. 18, where it more resembles two Gaussian distributions that are added.

There are also residual plots for the x coordinate that look like 2 gaussians where one is sharply peaked close to zero and one is really broad and further from zero. One explanation for this could be rotations; the chips can also rotate in respect to each other and this will cause one part of a chip to have a different residual than another part. Also rotations will cause a bigger change in the y coordinate than the x coordinate, since the chips are twice as big in the y direction than in the x direction. This can be seen in Fig. 19. Another part of the discrepancy is that there is a gap between the two chips which has not been properly included in the analysis of this dataset.

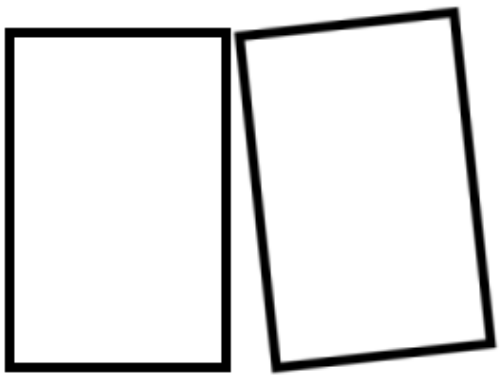


Figure 19: Schematic of a relative rotation of one chip in a single module.

5 Conclusions, discussion, and outlook

For the mTower I have measured the noise rate for different settings of VCASN and ITHR, which both control the threshold of charge needed to activate the pixels on the ALPIDE chip. I also determined the effect of VCASN and ITHR on cluster size using cosmic rays. From these cosmic rays I also selected events which produced straight lines and used them to get a rough measurement of the residuals of the chips. We have found that both VCASN and ITHR have a significant influence on noise rates, where VCASN is a more coarse determiner of the threshold and ITHR a fine one. We determined that noise rates for the recommended settings; VCASN 50 ITHR 51 has mean noise rate $(6.90 \pm 0.04) * 10^{-8}$ and VCASN 57 ITHR 51 has mean noise rate $(8.73 \pm 0.01) * 10^{-7}$. We also found that VCASN controls the threshold in coarse way and ITHR in a more fine way.
The mean cluster size for the setting VCASN 50 ITHR 51 was determined to be 3.34 ± 0.05 and to be primarily clusters of size 2, the cluster size for the setting VCASN 57 ITHR 51 was determined to be 4.10 ± 0.07 and primarily clusters of size 4. From the fact that we need clusters to better determine particle tracks but we need clusters to be small to not saturate the detector in case of large number of particles, one should find an optimum cluster size.

For the alignment we got first order corrections in the transnational directions x and y, we found that in the y direction they are much larger presumably due to rotations. In my code I tried to incorporate rotations by dividing up the chips into four quadrants and plotting the residuals for the quadrants which should give a translation that is constant for all quadrants and the remaining part should be the rotation. Sadly my code was bugged and all my residuals were mapped to a single quadrant for each chip. From debugging I could not resolve this, it already seemed to map the coordinates of the clusters to a single quadrant which was odd. Retracing the steps for determining the coordinates of the clusters didn't reveal where I went wrong. In the end I could have implemented the alignment code in another way to incorporate rotations better. I found an article and a presentation by Blobel[5], which explains that my fitting method is biased and can be seen as a first approximations to the true alignment. The bias can be seen in Fig. 16 where the best line fit and real particle track do not line up due to bias of the layer which is misaligned, we are biased towards the misalignment so should exclude these layers in the line fit. Thus better way of a fitting algorithm would be to say that there are two layers which are assumed as perfect which are then used to fit the straight line and the fitted line is then used to determine the residuals of the other layers. Another possibility is to determine the displacement of one layer by only using all the other layers in the straight line fit to remove bias.
But all methods using residuals have a bias[5], this bias can be avoided by using the alignment parameters in the parameters to determine the track. And then determine the parameters from all tracks instead of piece-wise determination, this will require a solution of a square matrix equal in size to the number of tracks. There is an algorithm called Millipede algorithm, which is implemented in C++ and the ROOT framework, that claims to be able to do this efficiently.

References

[1] A. P. de Haas et al., JINST **13**, P01014 (2018), 1708.05164.

[2] T. Peitzmann and M. H. Thoma, Phys. Rept. **364**, 175 (2002), hep-ph/0111114.

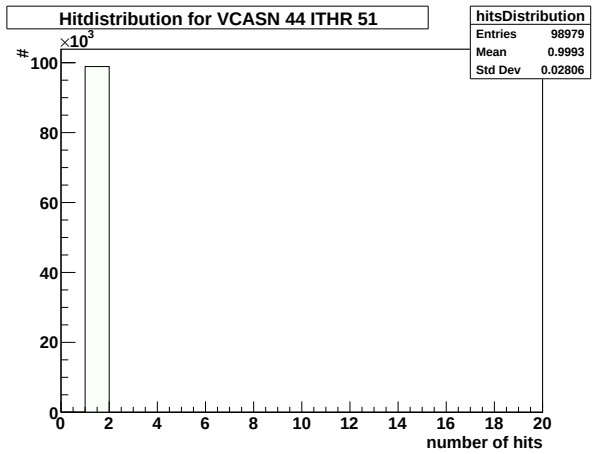
[3] F. Reidt, Ph.D. thesis, Heidelberg U. (2016-02-15), URL http://www.ub.uni-heidelberg.de/archiv/20648.

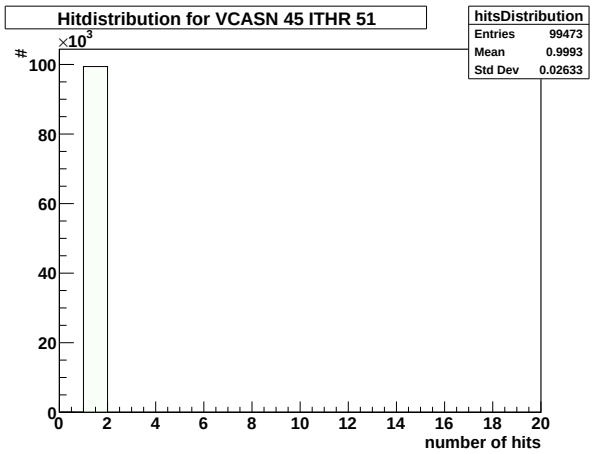
[4] T. P. A. Collaboration and e. a. Aab, Science **357**, 1266 (2017), ISSN 0036-8075, https://science.sciencemag.org/content/357/6357/1266.full.pdf, URL https://science.sciencemag.org/content/357/6357/1266.

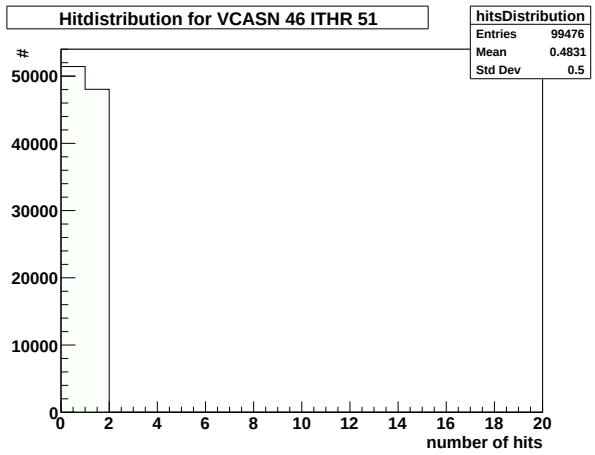
[5] V. Blobel, Nucl. Instrum. Meth. **A566**, 5 (2006).

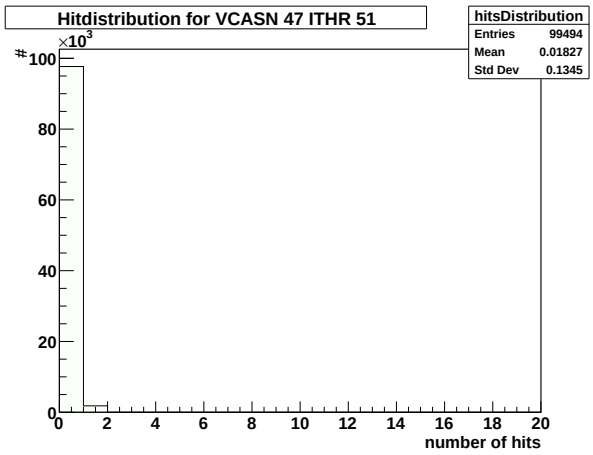
A Appendix

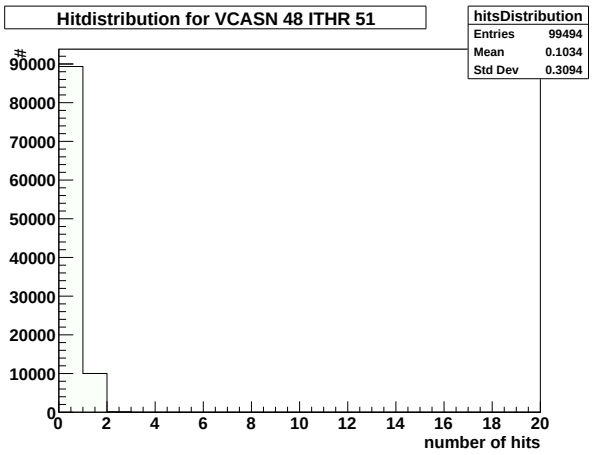
A.1 VCASN hitdistributions

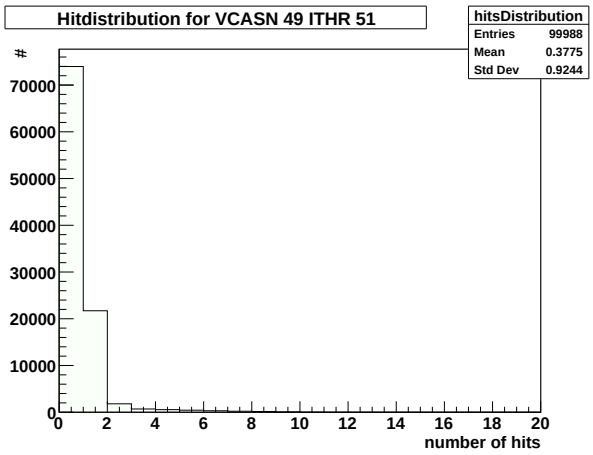


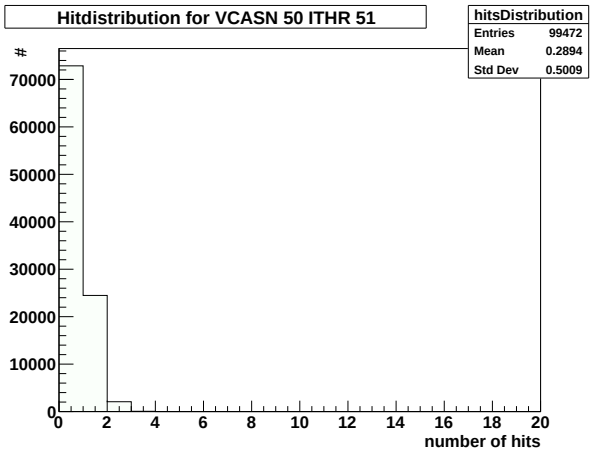


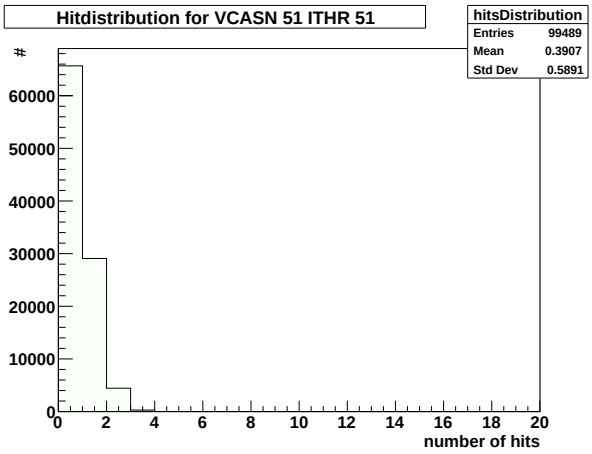


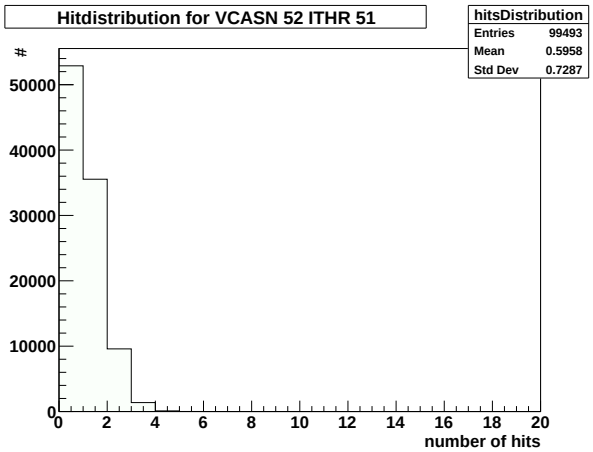


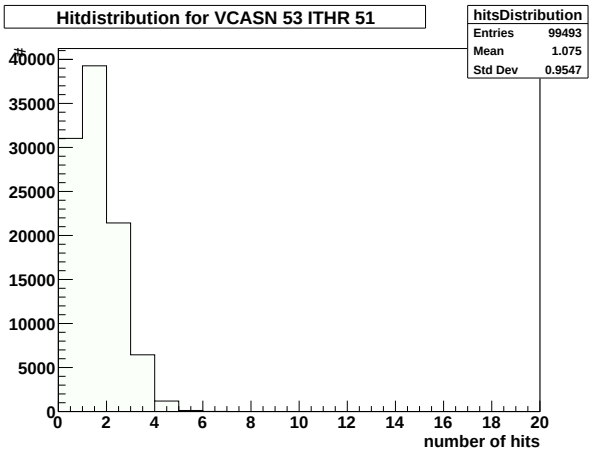


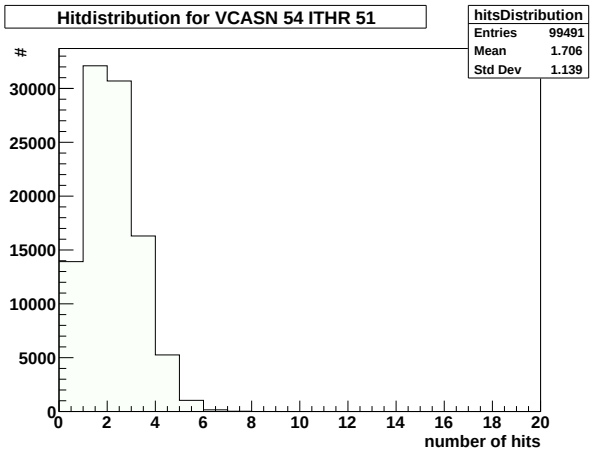


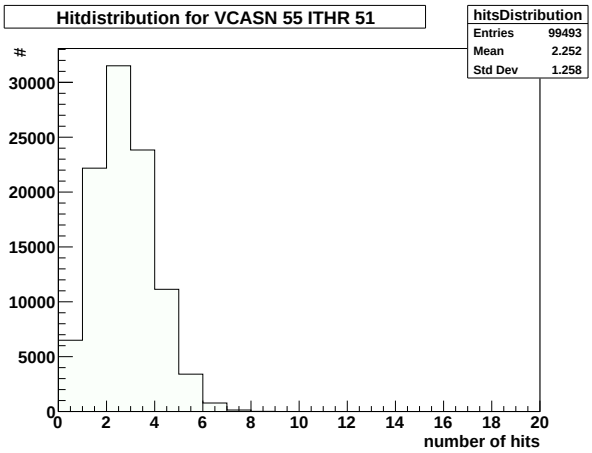

Hitdistribution for VCASN 50 ITHR 51
hitsDistribution
Entries 99472
Mean 0.2894
Std Dev 0.5009
#
number of hits

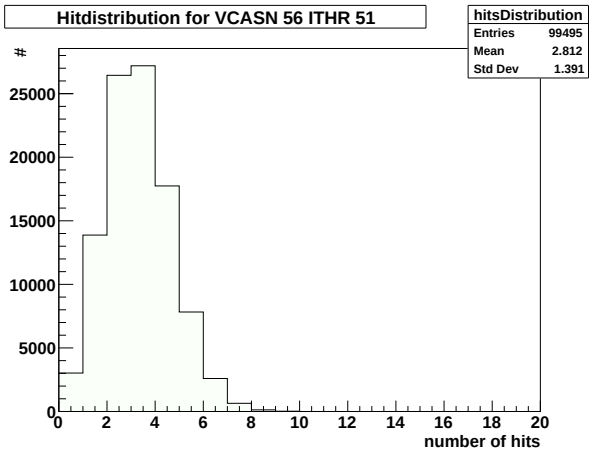

Hitdistribution for VCASN 51 ITHR 51
hitsDistribution
Entries 99489
Mean 0.3907
Std Dev 0.5891
#
number of hits

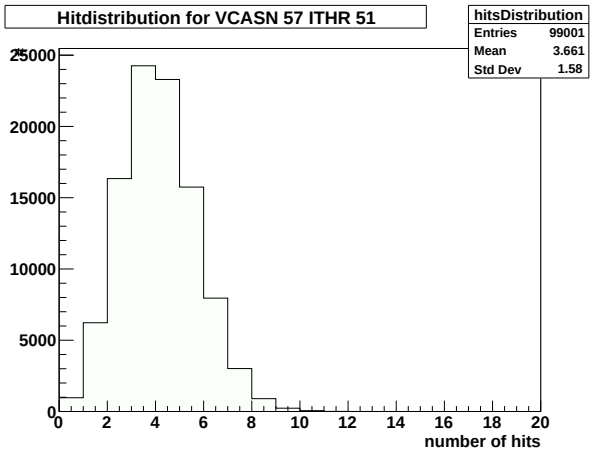

Hitdistribution for VCASN 52 ITHR 51
hitsDistribution
Entries 99493
Mean 0.5958
Std Dev 0.7287
#
number of hits

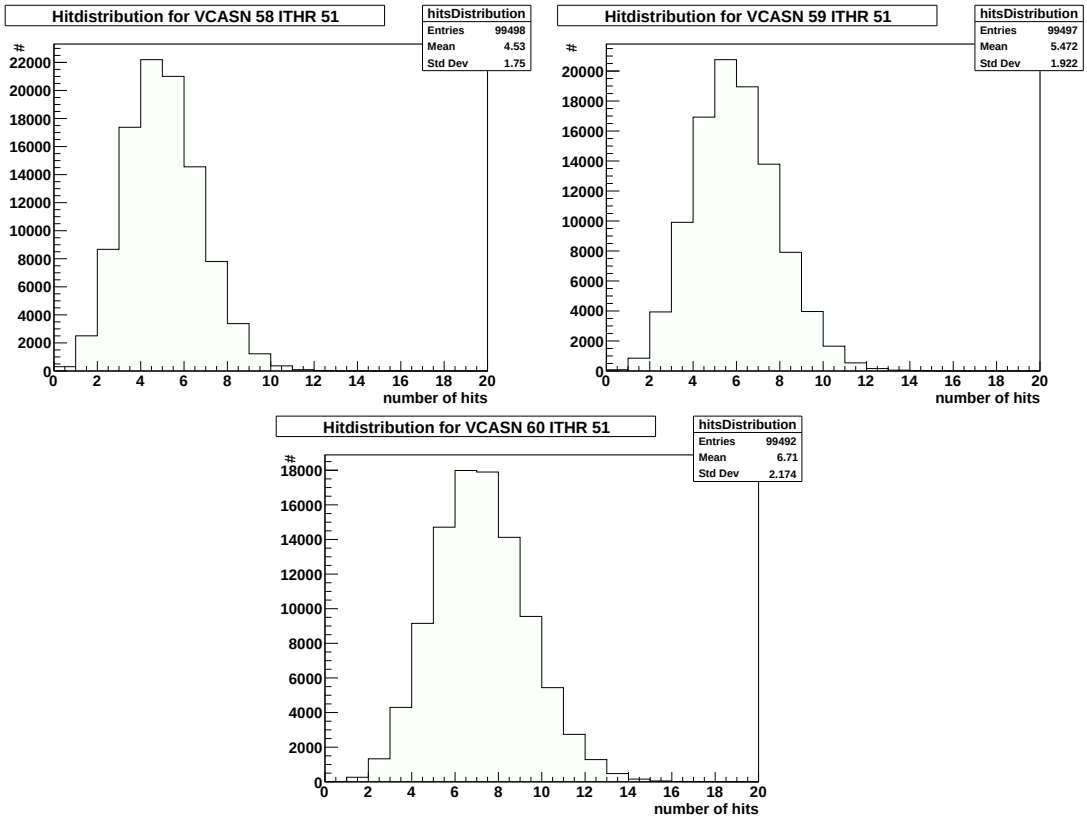

Hitdistribution for VCASN 53 ITHR 51
hitsDistribution
Entries 99493
Mean 1.075
Std Dev 0.9547
#
number of hits


Hitdistribution for VCASN 54 ITHR 51
hitsDistribution
Entries 99491
Mean 1.706
Std Dev 1.139
#
number of hits

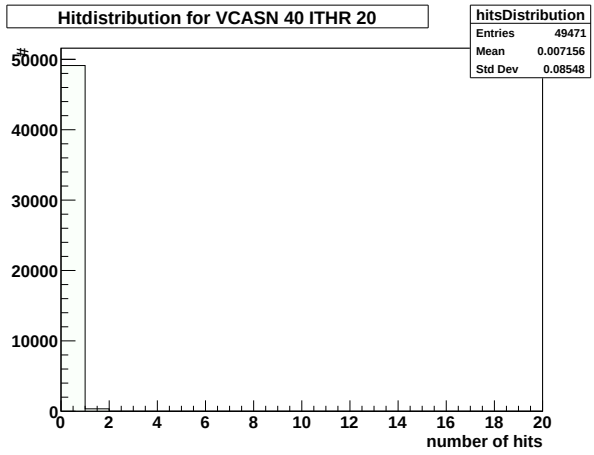

Hitdistribution for VCASN 55 ITHR 51
hitsDistribution
Entries 99493
Mean 2.252
Std Dev 1.258
#
number of hits

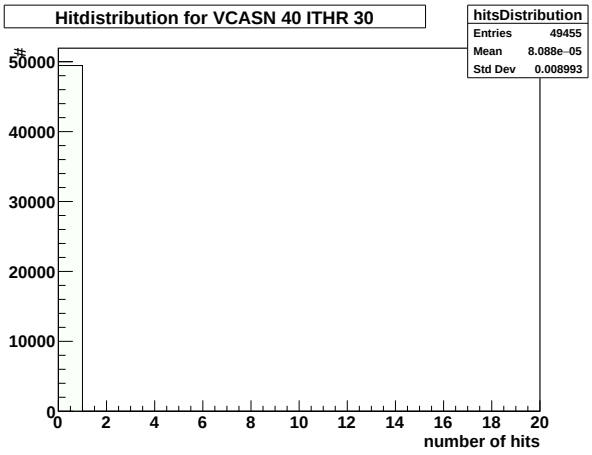

Hitdistribution for VCASN 56 ITHR 51
hitsDistribution
Entries 99495
Mean 2.812
Std Dev 1.391
#
number of hits

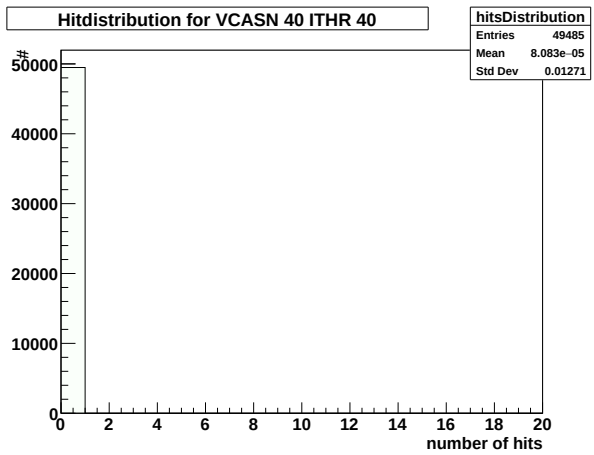

Hitdistribution for VCASN 57 ITHR 51
hitsDistribution
Entries 99001
Mean 3.661
Std Dev 1.58
#
number of hits

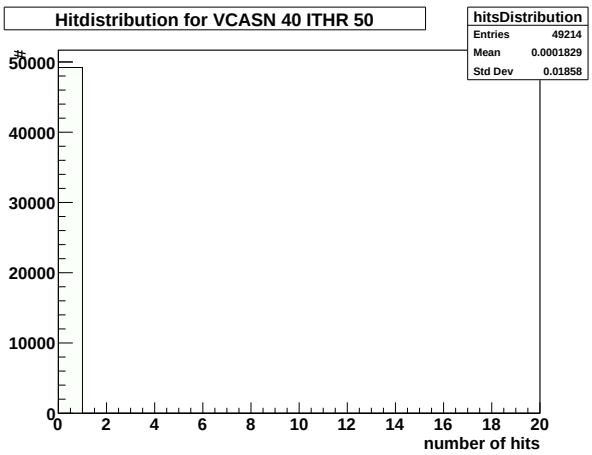

Hitdistribution for VCASN 58 ITHR 51
hitsDistribution
Entries 99498
Mean 4.53
Std Dev 1.75
#
number of hits
Hitdistribution for VCASN 59 ITHR 51
hitsDistribution
Entries 99497
Mean 5.472
Std Dev 1.922
#
number of hits
Hitdistribution for VCASN 60 ITHR 51
hitsDistribution
Entries 99492
Mean 6.71
Std Dev 2.174
#
number of hits

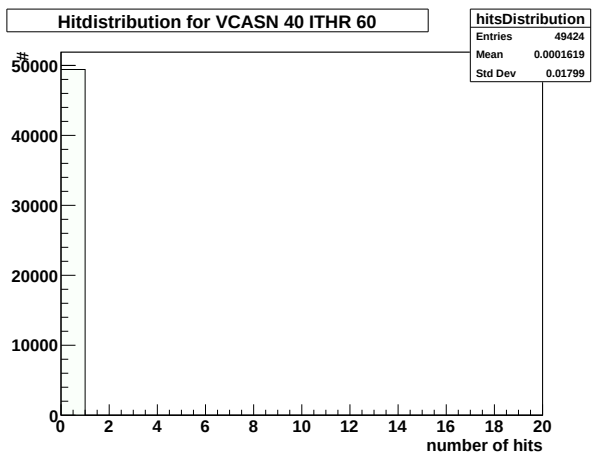
A.2 ITHR hitdistributions

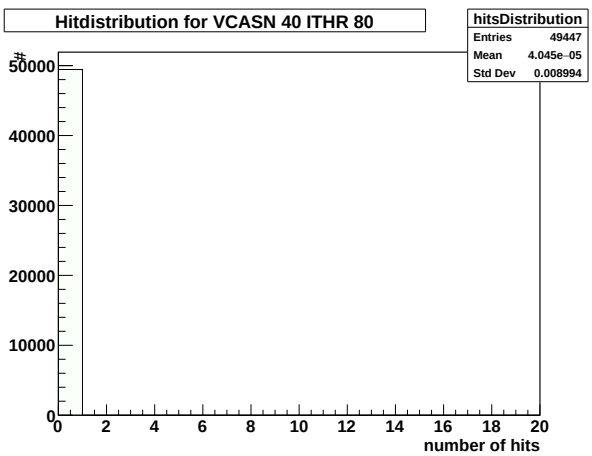


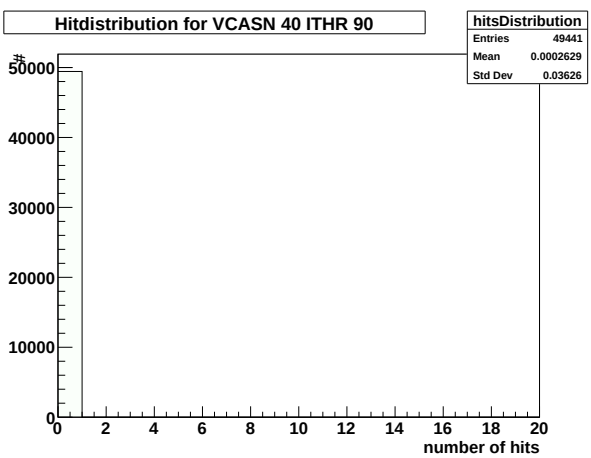


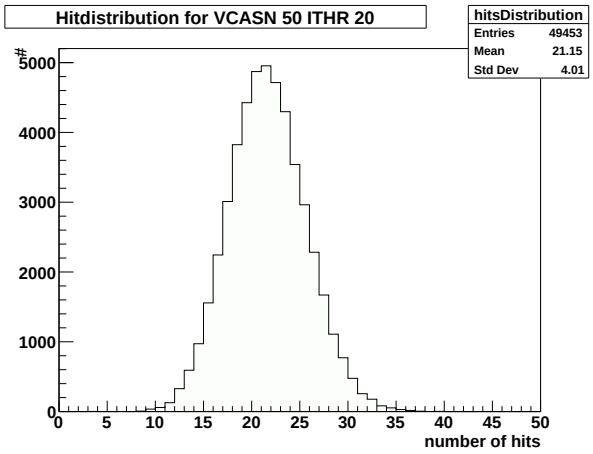


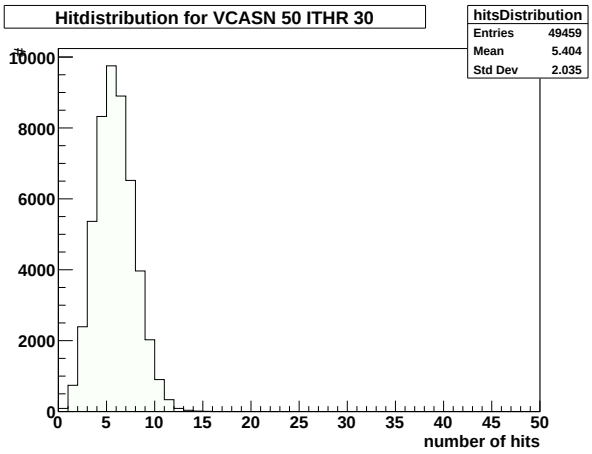


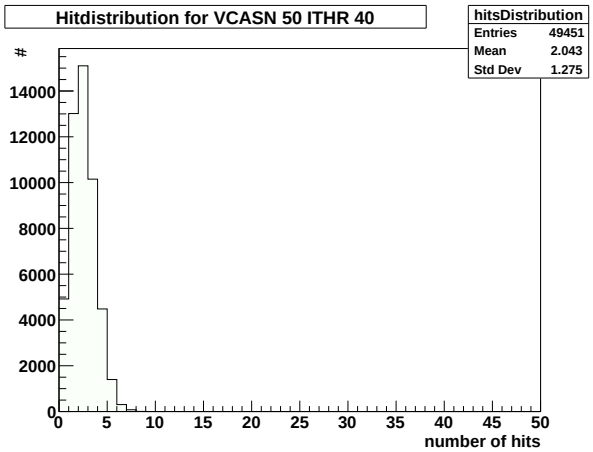


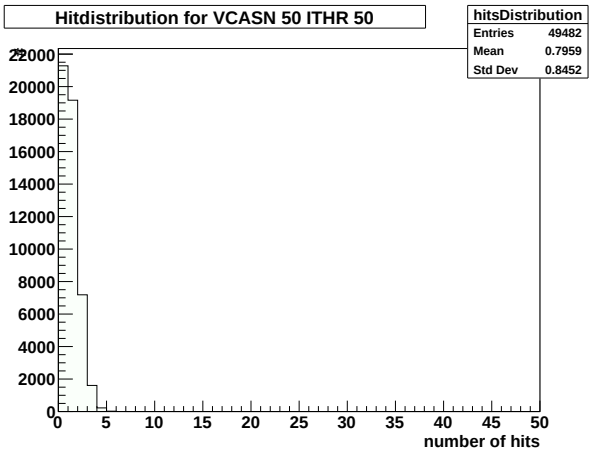


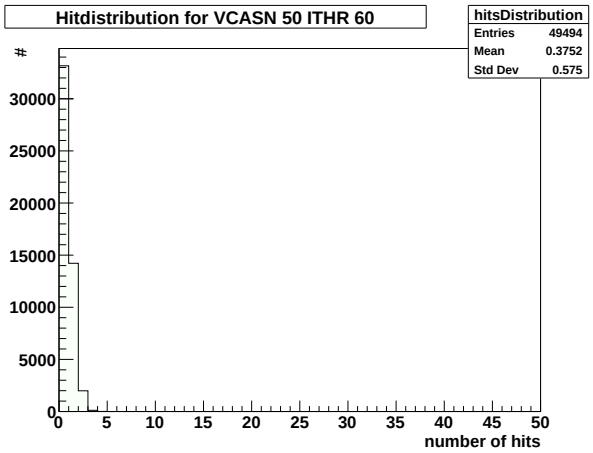


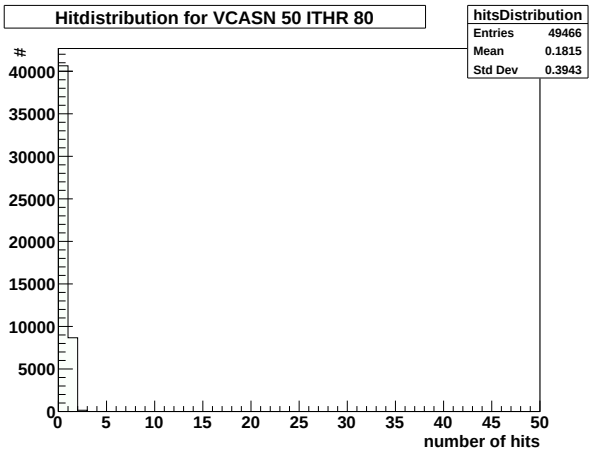

Hitdistribution for VCASN 50 ITHR 20
hitsDistribution
Entries 49453
Mean 21.15
Std Dev 4.01
#
number of hits

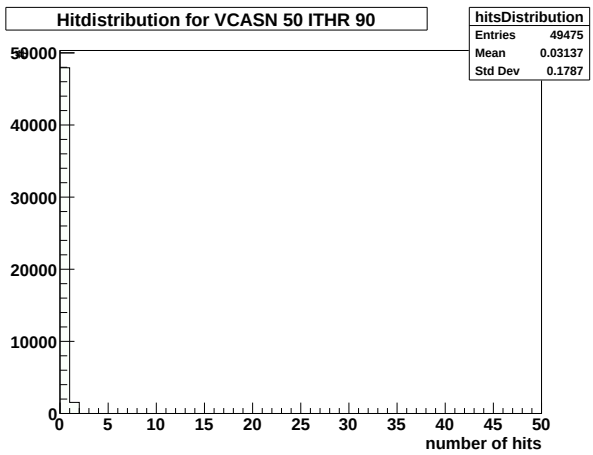

Hitdistribution for VCASN 50 ITHR 30
hitsDistribution
Entries 49459
Mean 5.404
Std Dev 2.035
#
number of hits

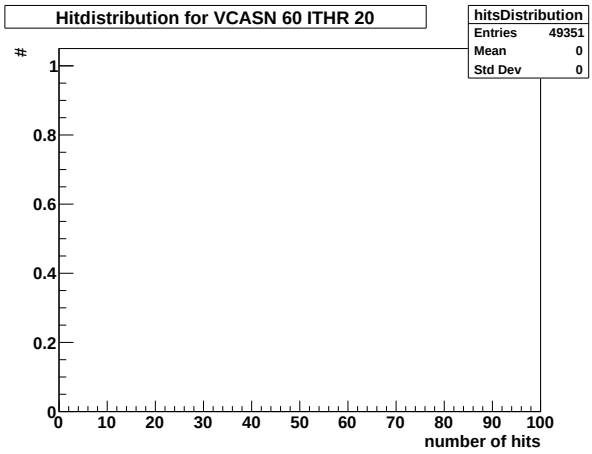

Hitdistribution for VCASN 50 ITHR 40
hitsDistribution
Entries 49451
Mean 2.043
Std Dev 1.275
#
number of hits

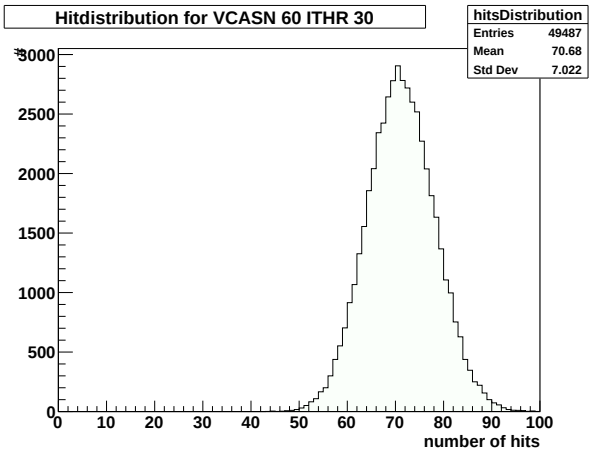

Hitdistribution for VCASN 50 ITHR 50
hitsDistribution
Entries 49482
Mean 0.7959
Std Dev 0.8452
#
number of hits

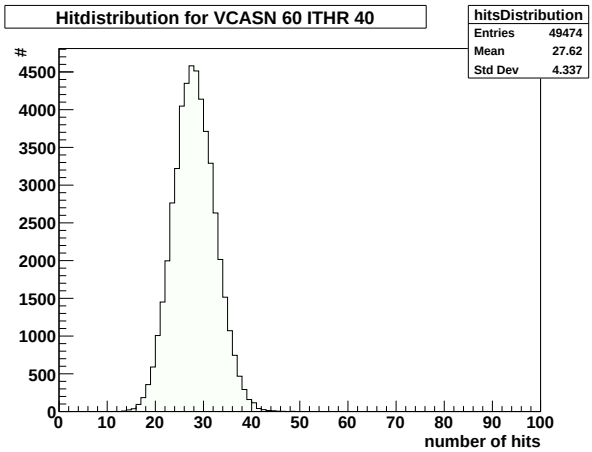

Hitdistribution for VCASN 50 ITHR 60
hitsDistribution
Entries 49494
Mean 0.3752
Std Dev 0.575
#
number of hits

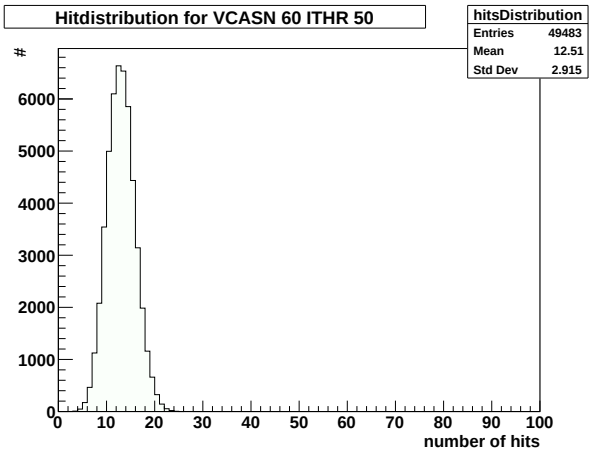

Hitdistribution for VCASN 50 ITHR 80
hitsDistribution
Entries 49466
Mean 0.1815
Std Dev 0.3943
#
number of hits

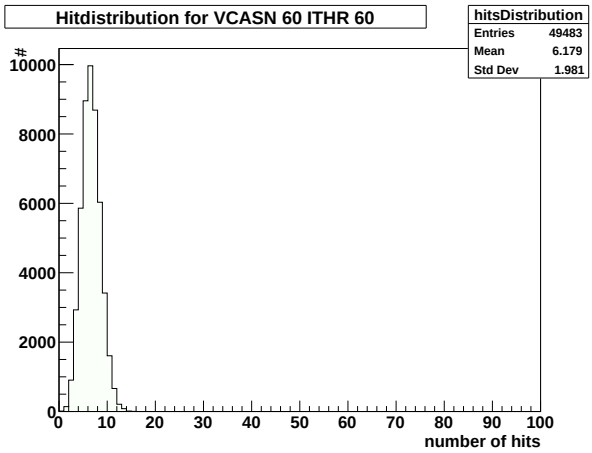

Hitdistribution for VCASN 50 ITHR 90
hitsDistribution
Entries 49475
Mean 0.03137
Std Dev 0.1787
number of hits

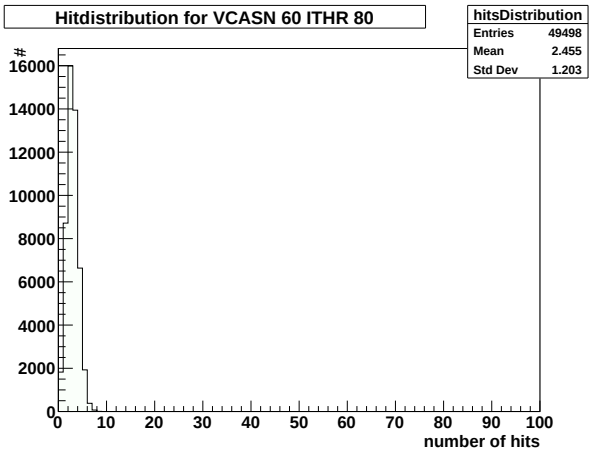

Hitdistribution for VCASN 60 ITHR 20
hitsDistribution
Entries 49351
Mean 0
Std Dev 0
#
number of hits

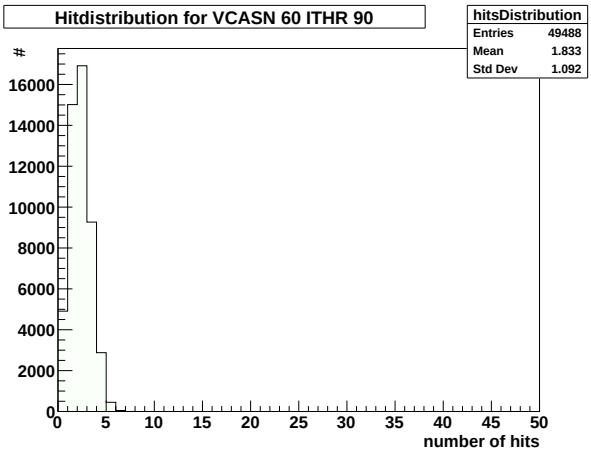

Hitdistribution for VCASN 60 ITHR 30
hitsDistribution
Entries 49487
Mean 70.68
Std Dev 7.022
#
number of hits


Hitdistribution for VCASN 60 ITHR 40
hitsDistribution
Entries 49474
Mean 27.62
Std Dev 4.337
#
number of hits

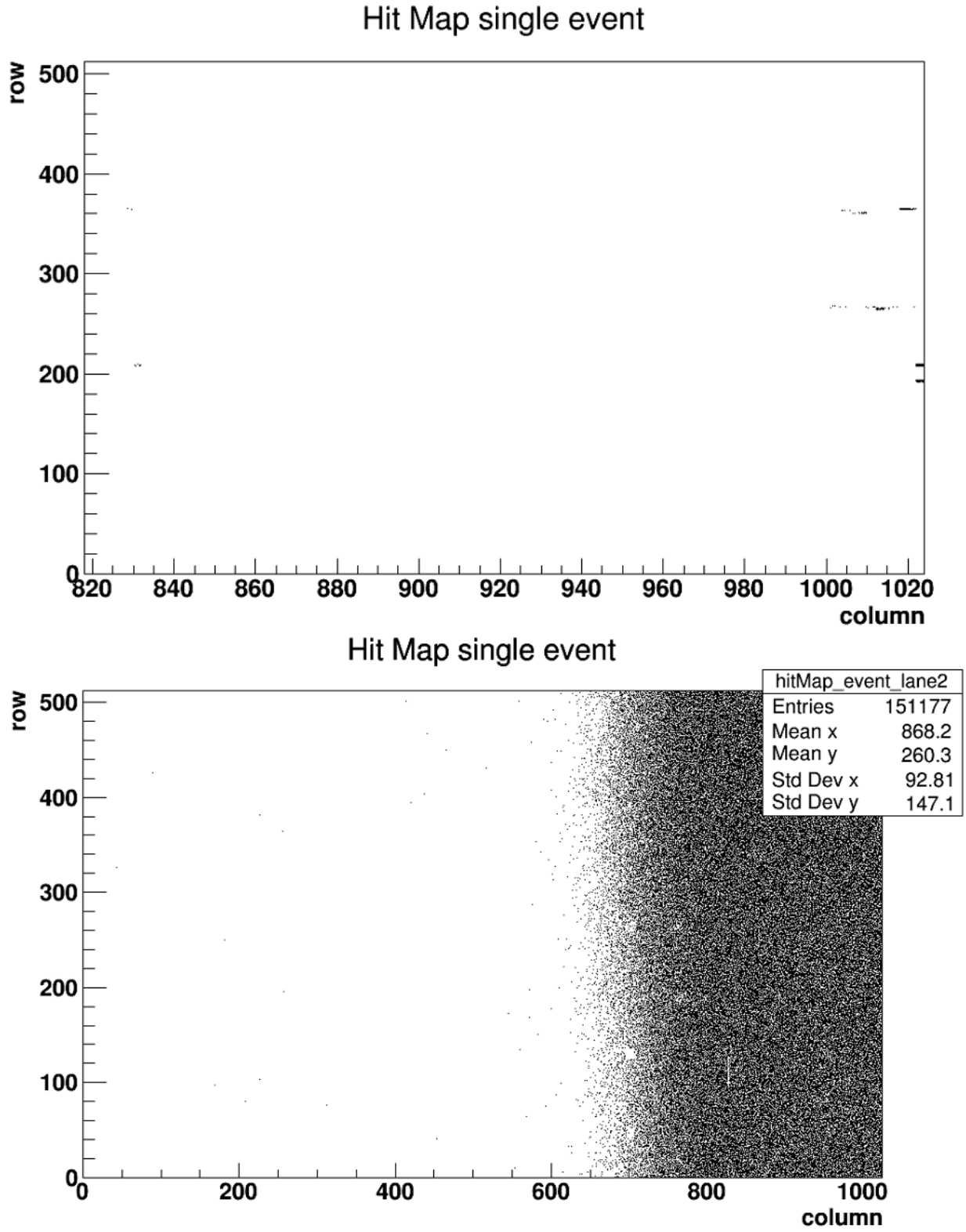

Hitdistribution for VCASN 60 ITHR 50
hitsDistribution
Entries 49483
Mean 12.51
Std Dev 2.915
#
number of hits


Hitdistribution for VCASN 60 ITHR 60
hitsDistribution
Entries 49483
Mean 6.179
Std Dev 1.981
#
number of hits

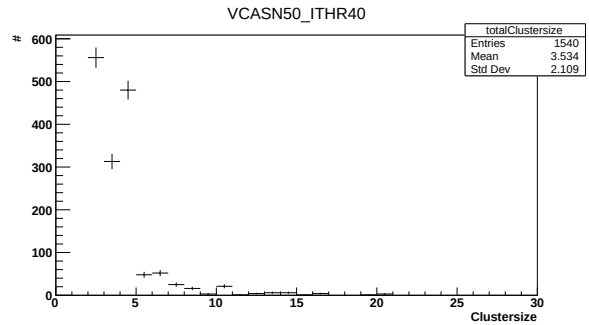

Hitdistribution for VCASN 60 ITHR 80
hitsDistribution
Entries 49498
Mean 2.455
Std Dev 1.203
#
number of hits

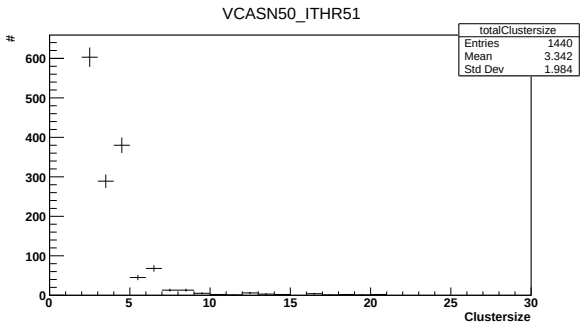

Hitdistribution for VCASN 60 ITHR 90
hitsDistribution
Entries 49488
Mean 1.833
Std Dev 1.092
#
number of hits

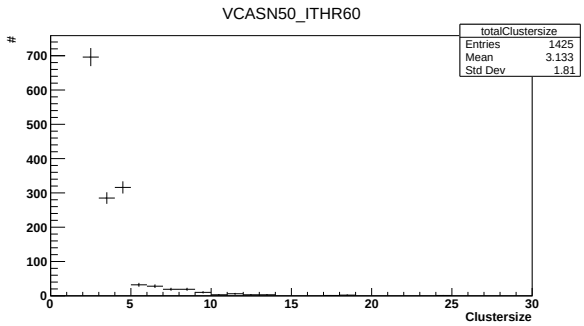
A.3 ITHR chip malfunctions

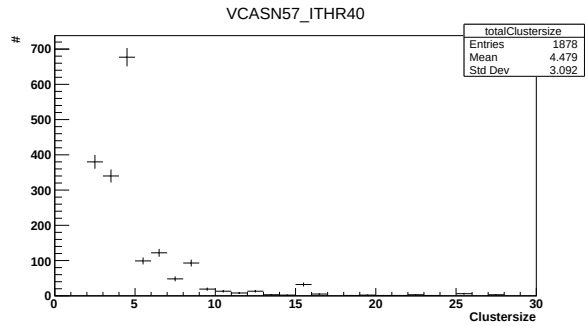


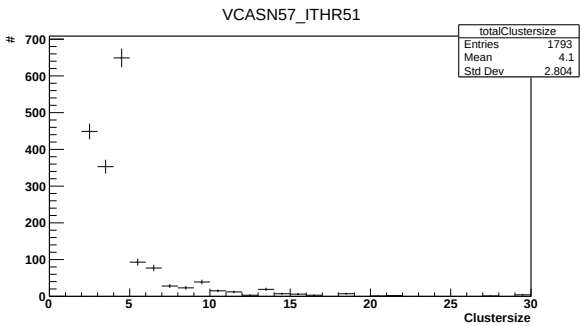
A.4 Clustersize

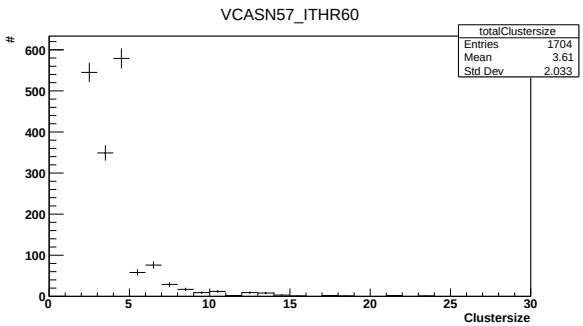




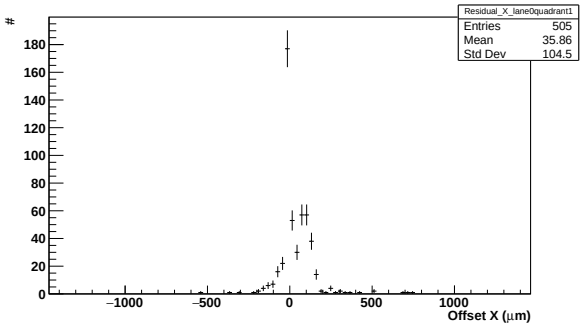


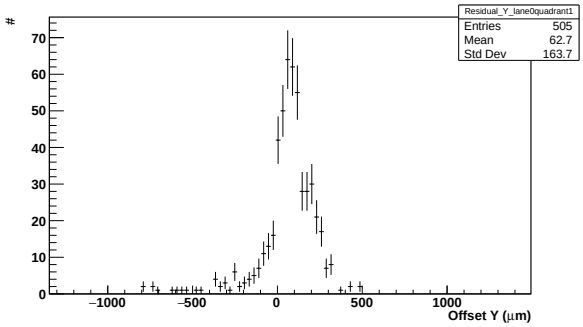


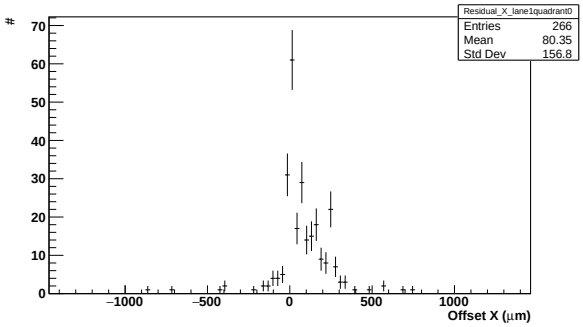


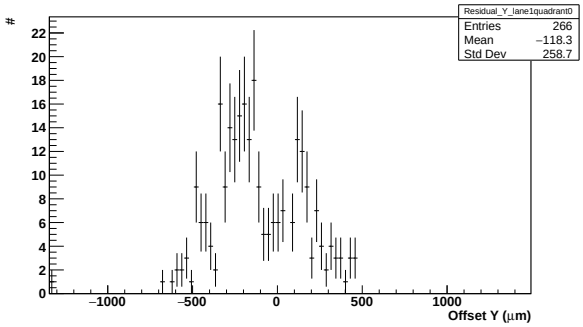


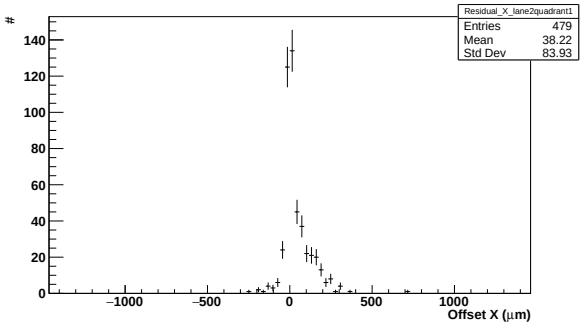
A.5 Alignment

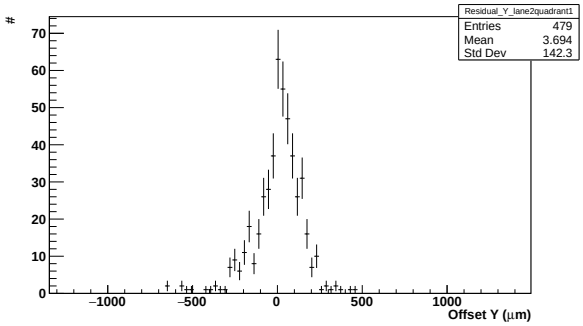


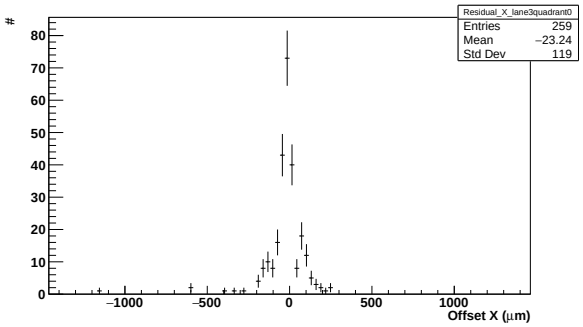


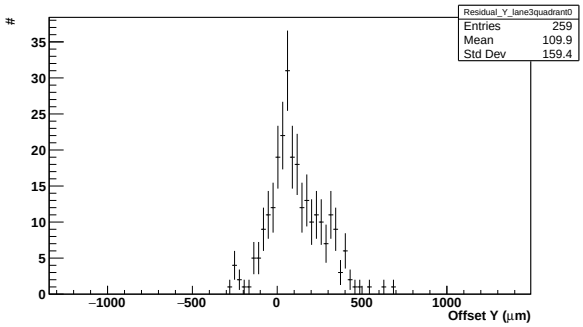


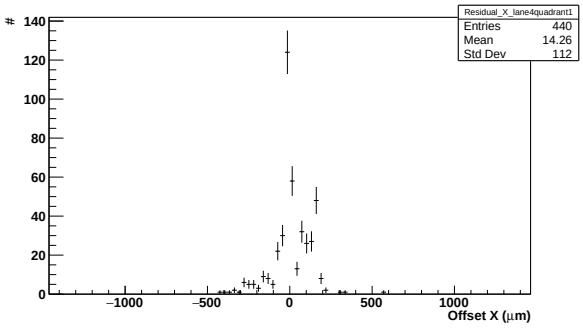


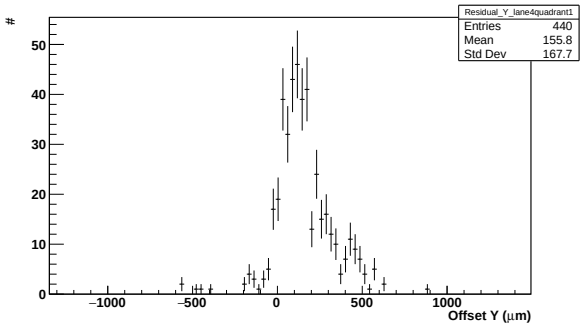


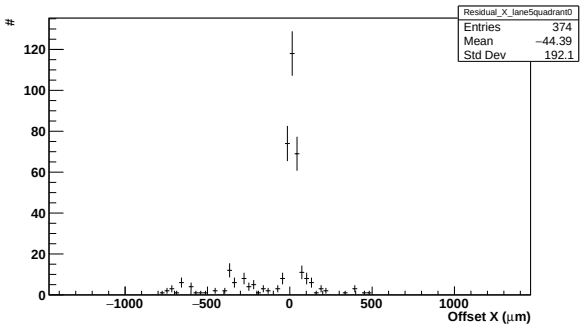


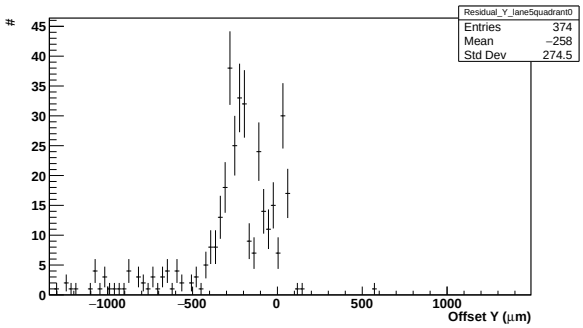


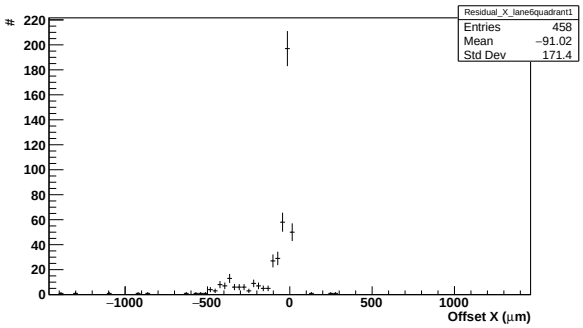


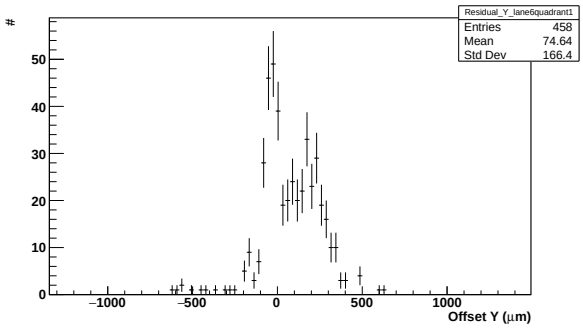

Residual_X_lane4quadrant1
Entries 440
Mean 14.26
Std Dev 112
#
Offset X (μm)

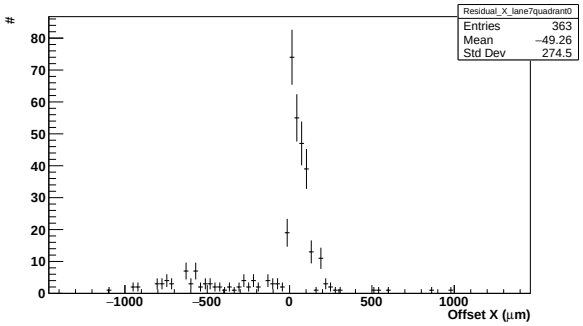

Residual_Y_lane4quadrant1
Entries 440
Mean 155.8
Std Dev 167.7
#
Offset Y (μm)

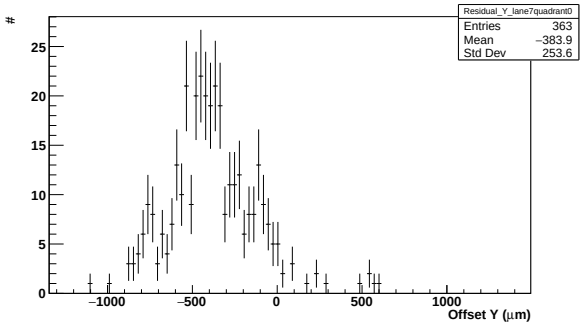

Residual_X_lane5quadrant0
Entries 374
Mean −44.39
Std Dev 192.1
#
Offset X (μm)

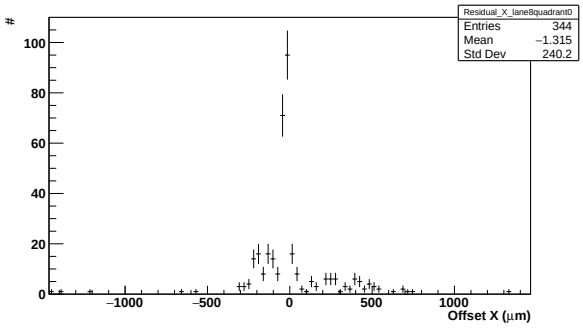

Residual_Y_lane5quadrant0
Entries 374
Mean −258
Std Dev 274.5
#
Offset Y (μm)

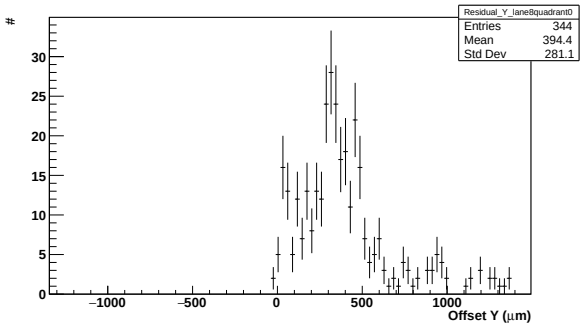

Residual_X_lane6quadrant1
Entries 458
Mean −91.02
Std Dev 171.4
#
Offset X (μm)

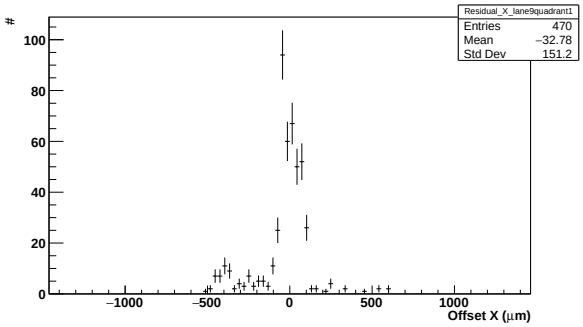

Residual_Y_lane6quadrant1
Entries 458
Mean 74.64
Std Dev 166.4
#
Offset Y (μm)

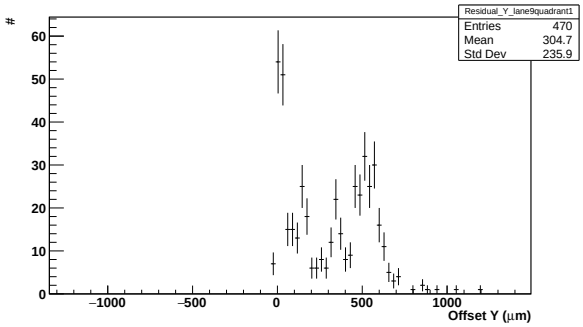

Residual_X_lane7quadrant0
Entries 363
Mean −49.26
Std Dev 274.5
#
Offset X (μm)

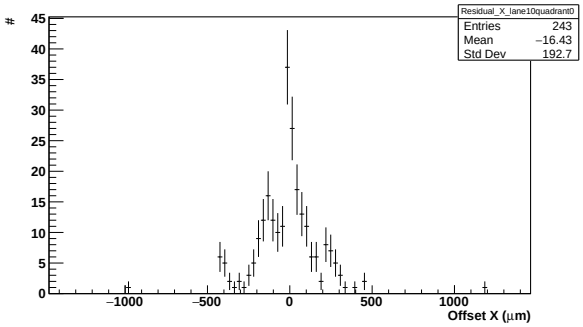

Residual_Y_lane7quadrant0
Entries 363
Mean −383.9
Std Dev 253.6
#
Offset Y (μm)

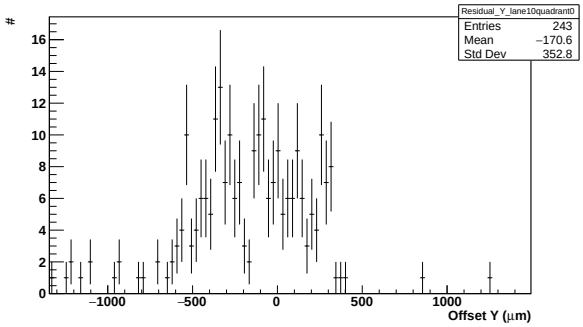

Residual_X_lane8quadrant0
Entries 344
Mean −1.315
Std Dev 240.2
#
Offset X (μm)

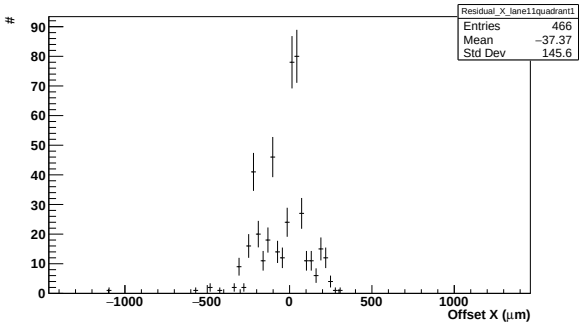

Residual_Y_lane8quadrant0
Entries 344
Mean 394.4
Std Dev 281.1
#
Offset Y (μm)

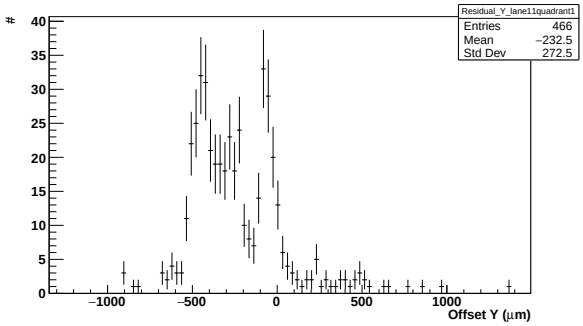

Residual_X_lane9quadrant1
Entries 470
Mean −32.78
Std Dev 151.2
#
Offset X (μm)

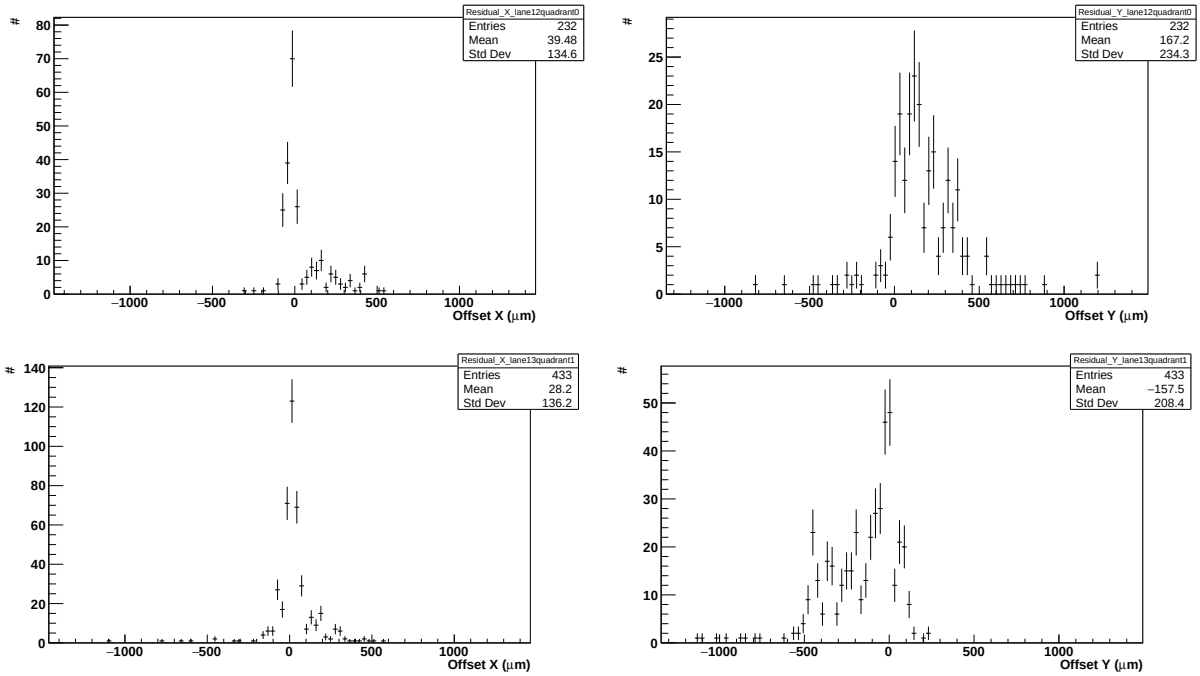

Residual_Y_lane9quadrant1
Entries 470
Mean 304.7
Std Dev 235.9
#
Offset Y (μm)


Residual_X_lane10quadrant0
Entries 243
Mean −16.43
Std Dev 192.7
#
Offset X (μm)


Residual_Y_lane10quadrant0
Entries 243
Mean −170.6
Std Dev 352.8
#
Offset Y (μm)


Residual_X_lane11quadrant1
Entries 466
Mean −37.37
Std Dev 145.6
#
Offset X (μm)


Residual_Y_lane11quadrant1
Entries 466
Mean −232.5
Std Dev 272.5
#
Offset Y (μm)



A.5.1 Alignment code (where to find)

```
//      Log in to the Quark cluster
//      go to directory of user 3876888

cd Code/                                    // go to code directory
cd Cosmic/                                  // go to code for Cosmics directory
emacs Alignment_Cosmic.C &                  // the alignment code
```