SLAC Testbeam Data Analysis: High Occupancy Tracking & FE-I4 Cluster Study

DESY Summer Student Project 2014

Martin Klassen, University of Heidelberg, Germany

September 27, 2014

Abstract

In this report a analysis of data taken from a high occupancy test beam at SLAC and cluster studies of the ATLAS FE-I4 readout module is presented. The set-up of the experiment and the different steps of the analysis are described.

The focus of this project was to investigate clusters with 2 or more tracks reaching into them and searching for a opportunity to split these clusters via their time over threshold value. This will be necessary for increasing the efficiency of reconstructing these tracks in the next run of the LHC with higher luminosity and the new inner most b layer which both leads to an increase of the possibility that two particles being closer together and therefore hit the same pixel or cluster.

Contents

1 Introduction

In this report first a short introduction to the telescope from a hardware perspective followed by a software discussion of the EUTelescope framework will be provided. Also the concept of Time over Threshold will be introduced before a short motivation of doing this analysis is given. Then there is a short description of the processors of the software added and every step is underlined with histograms from the data analysis. After this the results of the project are discussed and in the end a short conclusion is given.

1.1 The ACONITE Telescope

The ACONITE telescopes belongs to the EUDET project which is short for Detecor R & D towards the International Linear Collider, where electrons and positrons will be accelerated to a center of mass energy of up to 500GeV in the collisions. For this experiment detectors with a precise space resolution around 3 μm and a fast readout time of faster then 1kHz are required. The task of the telescope is to probe devices under test (DUTs). This DUTs can be for example new generations of pixel or strip detectors for the high energy physics detector as ATLAS or CMS and the telescope provides the possibility for track reconstruction for a particle going through the device. The telescope gives the opportunity to test these prototypes of detectors at a testbeam. With knowing the track through the measurements of the telescope and the reconstruction software framework the spatial resolution from the DUT can be measured very precisely.
Typical studies are efficiency, resolution in space, noise or readout time before and after radiation. The ACONITE is one of the 5 copies of the EUDET telescope. It consists of 6 parallel aligned planes of Monolithic Active Pixel Sensors (MAPS), so called MIMOSA26 sensors. This are divided into two arms with tree sensors each and in between there is a gap for placing the DUT. In this test run also a possible new part of the inner most detector for the next ATLAS upgrade the ATLAS FE-I4 was included in the set-up and put directly behind the second arm of the telescope. The entiere ACONITE telescope is shown in figure 1.

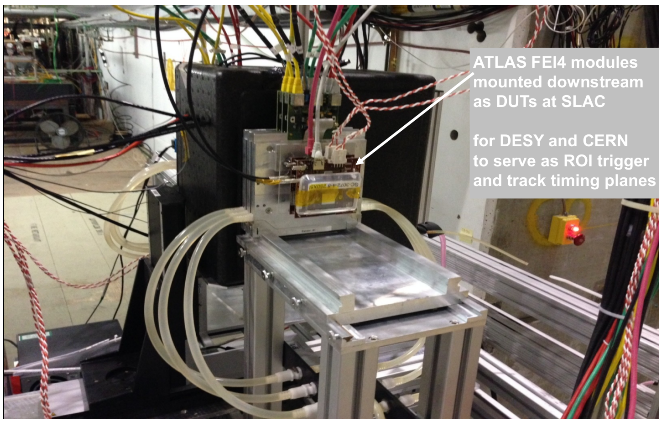


Figure 1: Set-up of ACONITE telescope with a ATLAS FE-I4 sensor at SLAC

1.1.1 Mimosa 26 Sensor

The most important part of the telescope which also provides the measurements of a particle going through the device are the 6 MIMOSA 26 silicon pixel sensors. The sensors and the readout palatines are fixed onto some aluminium planes. These can be moved freely on each arm separately for different studies, compare figure 2. The size of the MIMOSA sensors are 13.7 by 21.5 mm and cover an area of 2 cm^2 with 576 × 1152 pixels. These pixels are square pixels with a pitch of 18,4 μm, that means from one center to the center of the next pixel. The thickness of the sensors is around 50 μm of which only 14 μm is the active pixel layer. As more material is in the beam line as more multiple scattering will occur, which makes track reconstruction more difficult. The sensors can read out up to 10^6 particle-hits/cm^2/s which corresponds to a readout time around 115.2 μs or a frequency of around 8.68 kHz. Every pixel has a amplification and at the end of every column there is a discriminator followed by a analog digital converter with a zero-suppressed circuit. The data stream is then stored at two memory banks which allows simultaneously read and write.

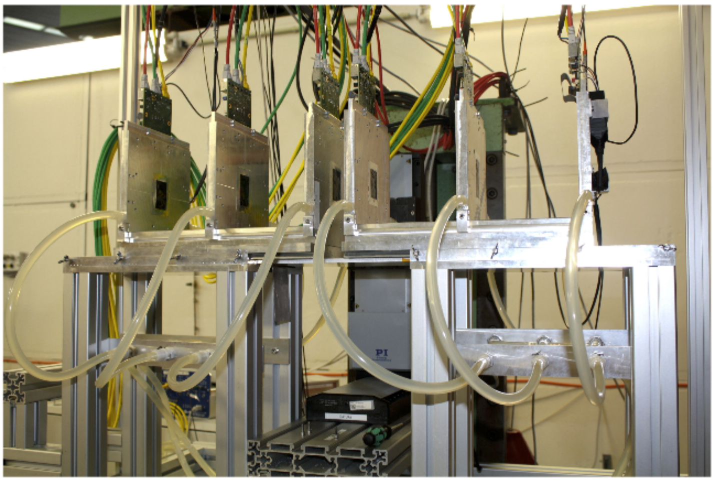


Figure 2: Side view of DATURA beam telescope [4]

1.1.2 Triggering

In front of the first and behind the last plane of the telescope there is every time one pair of crossed scintillator tubes which are read out via photomultiplier tubs (PMTs) which are used for triggering. They can be seen as the black boxes in figure 2 on the right side. If via a so called mask defined number of scintillators measure a signal, the pixels of all the telescope sensors and the DUT (if available) will be read out. Usually for starting the readout it is required that all 4 PMTs measured a signal which make it unlikely that they trigger without a particle going through. The signal from the PMTs is forwarded to the Trigger Logic Unit (TLU) which also adds an event number to every signal.

1.2 Atlas FE-I4

The ATLAS FE-I4 was developed as a new Front-End for the upgrade of the ATLAS detector for dealing high the high occupancy caused by the increase of the LHC luminosity. Other motivations are a decrease in material and a lower power consumption. The FE-I4 will be used in the Insertable B-Layer (IBL) upgrade and might be inserted as the outer layers of the detector after the Super-LHC upgrade.
It consists of a pixel array of 80 × 336 pixels. The pixel size is $250 \times 50\ \mu m^2$ which gives a decent spacial resolution. The readout rate for the FE-I4 is about 400×10^6 particle-hits/cm^2/s which is 400 times higher then for the MIMOSA planes. Due to the higher occupancy also the time over threshold bit size had to be decreased to 4 bits in comparison to 8 bits of the FE-I3 which leads to many energies belonging to the same ToT value.

1.3 Time over Threshhold (ToT)

Time over Threshold is a tool for calibration of the IBL for the energy deposit in a pixel by the particle going trough. The Time-over-Threshold is proportional to the induced charge. For the ATLAS FE-I4 every pixel is calibrated. For doing the calibration an external Pulser is put parallel to the real sensor readout which can inject a special amount of charge over to capacities. The signal both from the sensor as well as from the pulser will be shaped by the charge amplifier and forwarded to a digitised discriminator. In the end it will be passed to a digital read-out. The ToT value is the time the charge of the amplifier output is over a certain threshold. So if more charge is produced by a passing particle more charge will be collected by the sensor and brought to the amplifier and the signal will stay longer about the threshold, figure 3. This clarifies the proportional relation between the ToT and the amount off charge which is produced.

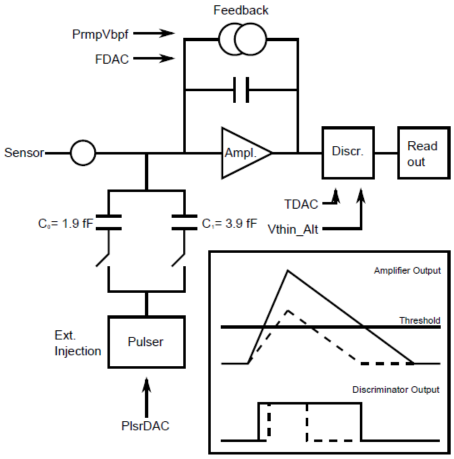


Figure 3: scheme for Time over Threshold [1]

The relation between ToT and charge is not linear so a calibration curve is needed to reconstruct the charge for example for clustering. For reducing the number of calibration constants, the calibration curves are derived for each front-end and not pixel-by-pixel wise. An inverse fit function is then used in the analysis which behaves well under inversion in the required range. Considered fit functions are :

$$< \overline{ToT} >= p_0 \frac{p_1 + Q}{p_2 + Q} \tag{1}$$

$$< \overline{ToT} >= p_0 + \frac{p_1}{p_2 + Q} \tag{2}$$

$$< \overline{ToT} >= p_0 + p_1 Q + p_2 Q^2 \tag{3}$$

For the 4 bits there are 16 possible ToT values:

ToT code	0-12	13	14	15
ToT [25ns]	ToT code +1	>13	delayed hit	no hit

- $ToT_{Code} = 0$ corresponds to ToT of 25 ns
- $ToT_{Code} = 15$ corresponds to no hit
- $ToT_{Code} = 13$ includes all large signals
→ Saturation for large charges due to ToT design
- $ToT_{Code} = 14$ is notification of delayed hit

2 Motivation

With the upgrade of the ATLAS innermost detector, the increase in luminosity, hadron boost and decrease in beam-radius of the LHC will resolve in particles being closer together on the IBL layer. That means it will get more likely that two tracks hit the same cluster or even the same pixel at the inner detectors. Reconstructing tracks of b-jets at high occupancy with IBL at 14 TeV will guide to a lower number of reconstructed events. With splitting the clusters with several tracks passing through via the time over threshold the number of found tracks should increase like shown in figure 4.

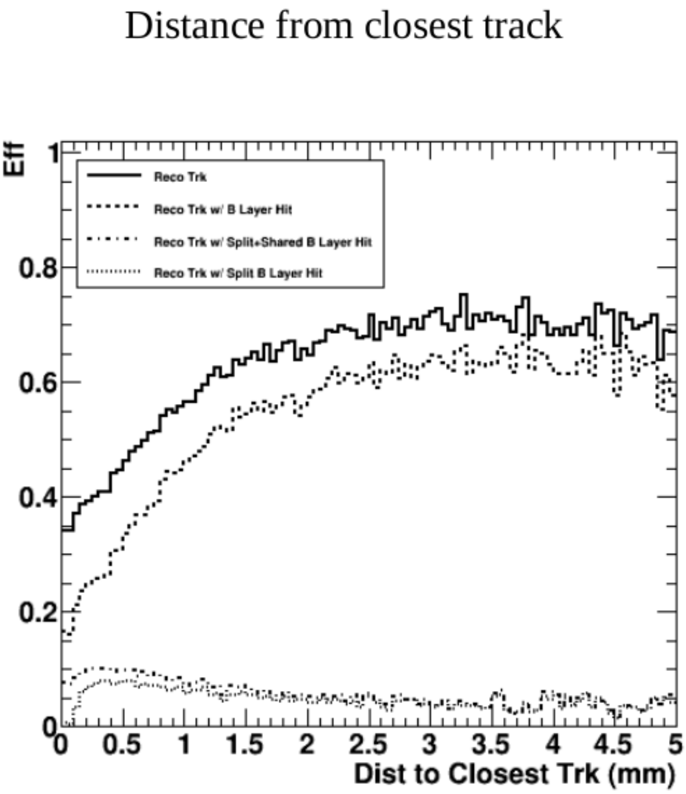


Figure 4: Simulation of ATLAS track reconstruction efficiency for distance from closest track; M. Battaglia UCSC and CERN

There the efficiency of reconstructed tracks drops for all reconstruction methods with decreasing distance between a track and the closest track next to this one. The effects starts at a minimal distance of 2,5 mm to the next track. The reconstruction possibility furthermore decreases stronger for tracks with a B layer hit. If the clusters of tracks with 2 tracks going into it can be split the track reconstruction can become about 50 % more efficient, for close tracks that means from around 17 % to 35 %. That shows that having a method to split clusters with several tracks and using them more then once in the track finding process is a good approach for getting higher statistics for example in the b-tagging approach. There will be a Testbeam in Oktober at SPS at CERN for doing further studies on this procedure.

3 Data Analysis with the EUTelescope framework

The EUTelescope framework shown at figure 5 is used for analysis and reconstruction of data from the beam telescopes. The goal of the software is to convert raw measured data from the telescopes to objects like tracks crossing the telescope which can be used for analysis first on the telescope itself and second on a device under test inserted into the telescope.

For these analysis the software is split into small processors grouped in the event processor Modular Analysis & Reconstruction for the LINear collider (Marlin). These provides the opportunity to run huge analysis via Grid computing and to do a step-by-step analysis. That means every task is implemented in a separate processor which is called then by Marlin. With steering templates the processors can give parameters back to the user, who can also configure these parameters during run time via the templates which have a Extensible Markup Language (XML) format.

The other main elements of the framework are the Liniear Collider I&O The (LCIO) data model and the Geometry API for Reconstruction (GEAR) markup language and the RAIDA processor. Where the last one is the implementation of ROOT into the Abstract Interface for Data Analysis (AIDA), the so called RAIDA. This processor is called by Marlin after every run and creates histograms and during the event process information are added. After the histograms are created they are accessible over the ROOT interface. Most of the data like particle tracks or measured hits is saved into so called EUTelFitTuples which are basically a ROOT tree what allows a easy inspection of all events.

The main input for the hole analysis chain is the LCIO ouptput file produced by the Data AcQuisition (DAQ). This contains the pixel raw data which are zero suppressed for the telescope, that means only if a pixel fires the data is forwarded and saved. It is consistent with most of the used programming languages in particle physics like C++, Java or Fortran and designed to be generally used in detector research and development. The format is event-based so that all data belonging to one trigger event and the underlying detector readout are stored together and accessible via the event number. After the track reconstruction process the hits and tracks are stored in two collections, TrackHit and Track collection and can be forwarded via the steering templates to the processors for the analysis.

In the Telescope Geometry Description the GEAR file the physical set-up of the telescope is saved. This makes it possible to run the entire processor chain for different detector geometries for examples the distance or angels between the Mimosa planes, the position of the Atlas FE-I4 or the pixel pitches. That makes a transformation from local hit coordinates left behind on the planes by a track going through the telescope to the global coordinates from the telescope frame possible.

Beside the pixel raw data other data like the pedestal and noise are needed for calibration or the alignment constants for finding the tracks. EUTelescope has several track reconstruction algorithms implemented.This analysis is based on DAF-Fitter(Deterministic Annealing Filter) other options would be GBL(general broken lines) or a straight fitter.

DAF-Fitter creates tracks out of hits belonging to a track over the highest probability of a group of clusters building this track.

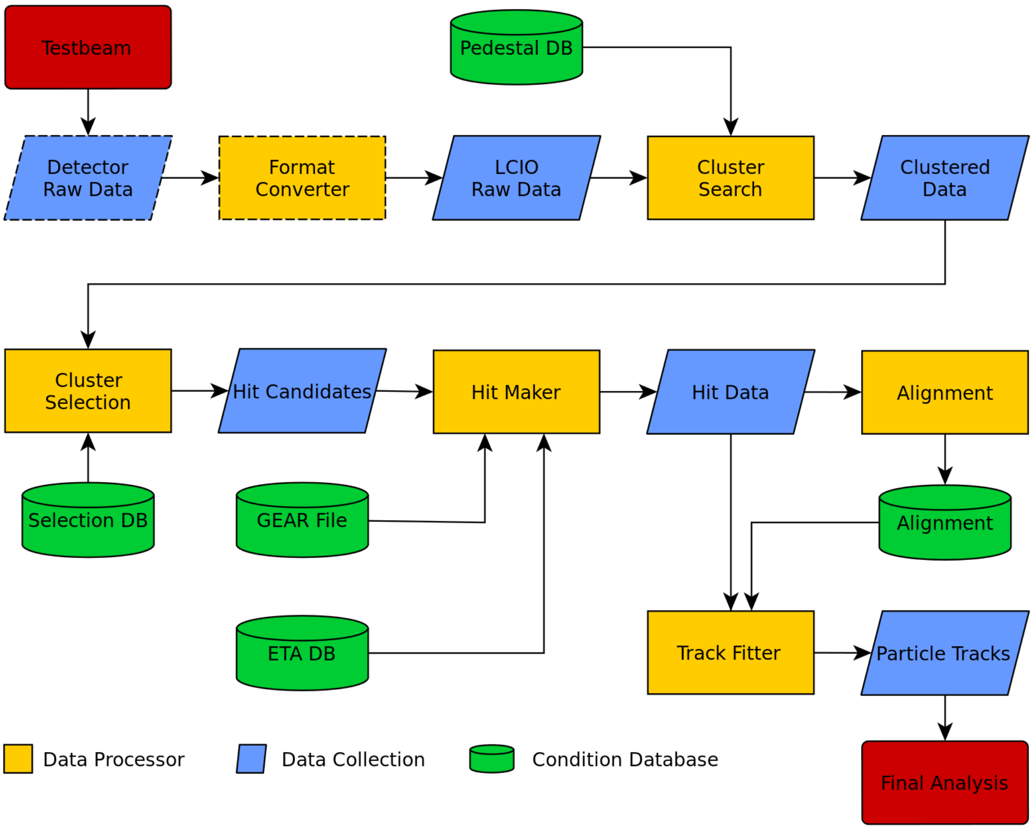


Figure 5: Scheme of the EUTelescope framework; http://eutelescope.web.cern.ch/content/about-eutelescope

3.1 Testbeam

For executing the studies of new detectors a decent environment is needed. For doing the precise measurements of space resolution or time the telescopes are build. To get a particle beam, a testbeam facility is needed. Typical places for this are the testbeam at DESY II, at CERN the PS and SPS or the data which is analysed in this report is taken at SLAC in 2014. The beam energy was about 12 TeV and per event which is defined trough one trigger signal, a event multiplicity of 80 to 90 clusters was reached. So in every event there were around 90 clusters at every plane hit by several tracks which in the further studies should be reconstructed again. In figure 6 for the second Mimosa plane which is refered to as plane 1, because counting starts at 0 the event multiplicity is shown for all events taken in this run with the same setup. All Mimosa planes are in line with a distance of 15 cm between each other and the Atlas FE-I4 modul is located directly behind the 6 Mimosa planes beam downwards.

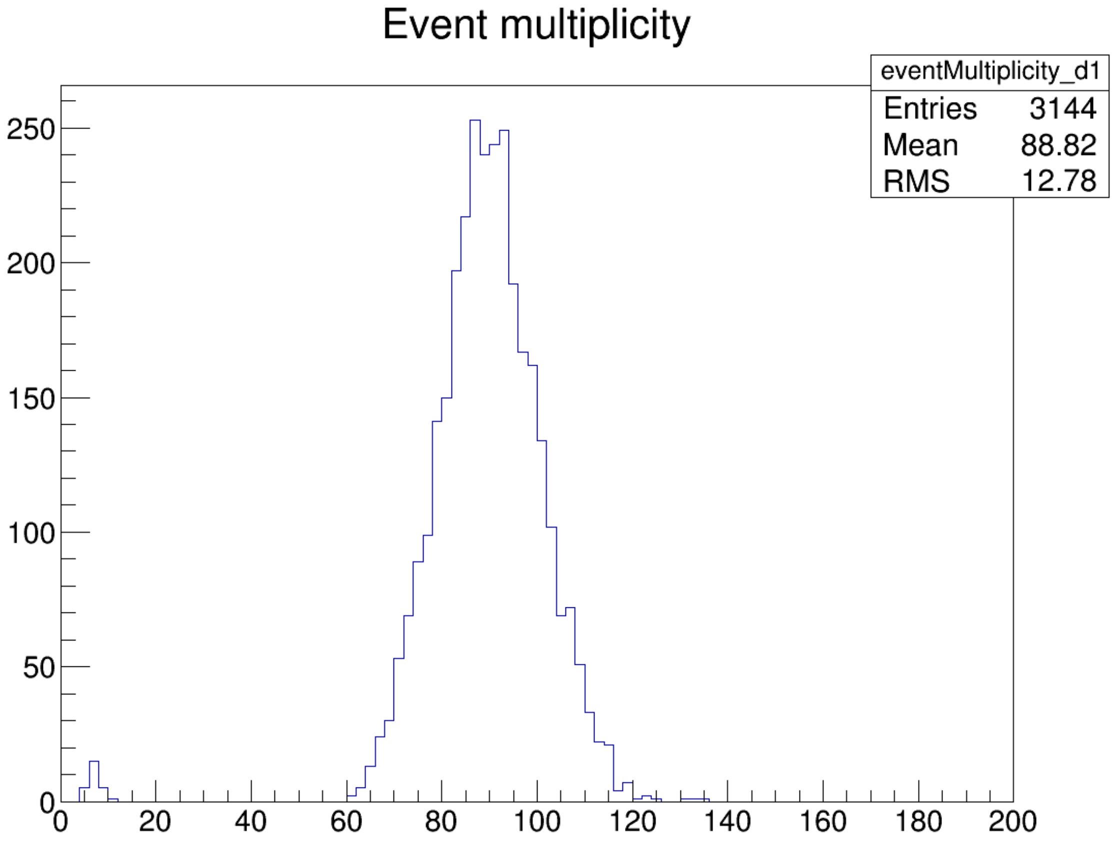


Figure 6: Event multiplicity at plane 1 of the telescope run 000295 from SLAC

3.2 Converter

In this first step of the data analysis the raw data from the detector like fired pixels forwarded by the DAQ software is converted into the LICO format, developed for linear collider studies. Furthermore hot pixels will be masked and in the following steps on no longer used for studies. Hot pixel means that this pixels fire at a very high frequency so that it is unlikely that they measured a real particle going through the detector but they might be broken for example.

3.3 Clustering

This step consists in general of two steps:

- **Cluster Search:** Particles hitting the Mimosa sensors often do not hit just one pixel they often hit also pixels in the neighbourhood and in this analysis step these are grouped together into a so called cluster. In this algorithm for every fired pixel the adjacent pixels are checked if they were activated as well. The cluster size is anti proportional to the pixel size itself and depending on the incident angle Θ as shown in figure 7.

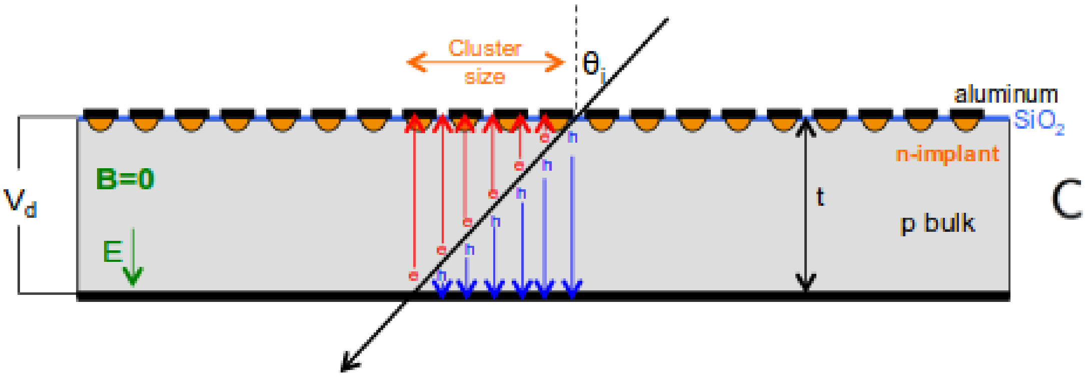
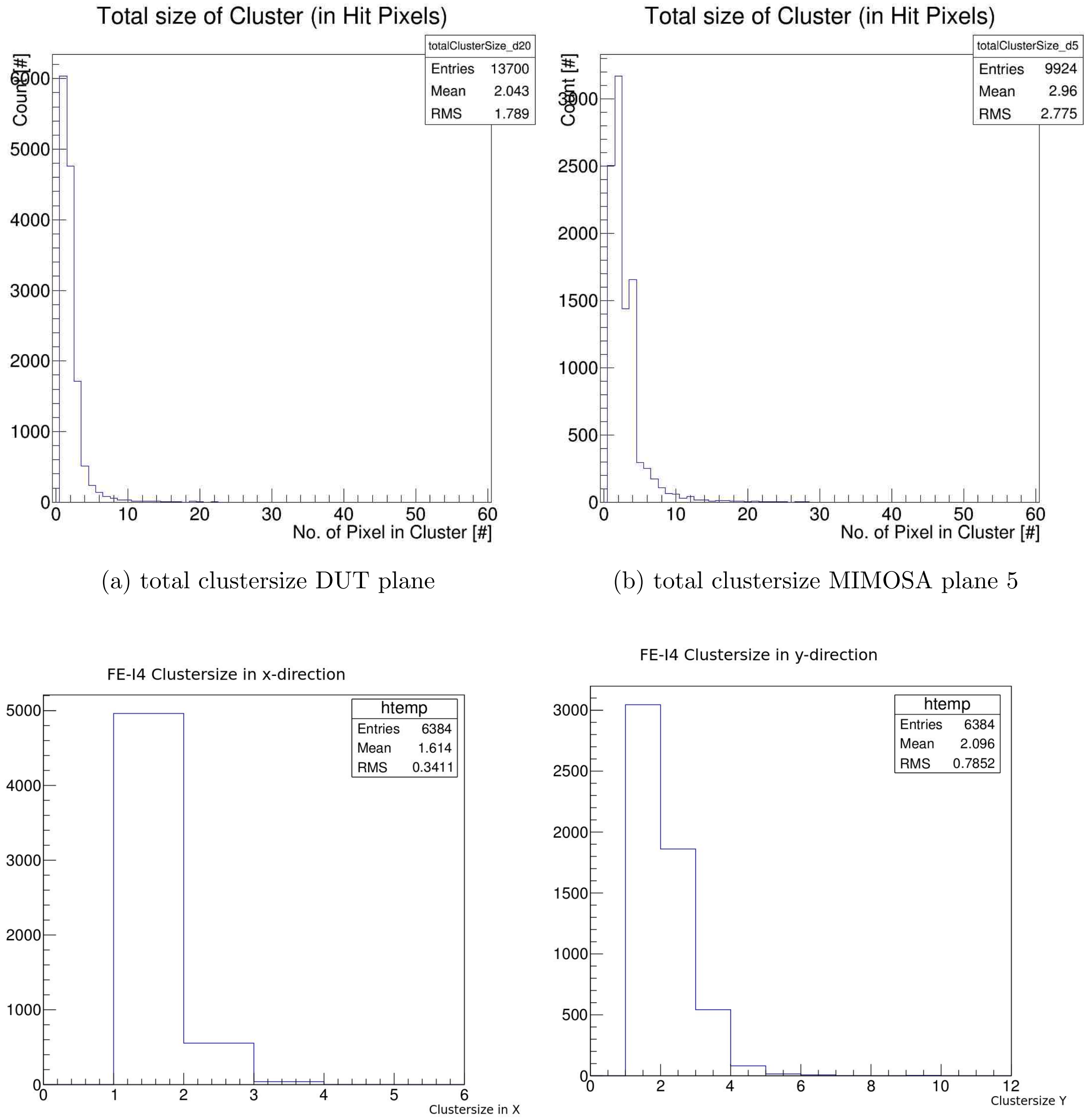


Figure 7: charged particle going through a silicon strip detector with an incident angle Θ_i without magnetic field; [3]

In figure 8 the cluster size is shown for the Mimosa26 sensors and the Atlas FE-I4. First of all the average size of a cluster is higher for the telescope set-up then the FE-I4 detector, compare 8a & 8b. This can be explained by the fact that the pixel size of the MIMOSA sensors is 18,4 μm in x and y direction. But for the Atlas FE-I4 we have a pixel size of $50 \times 250\,\mu$m which makes it more likely that a particle going through the detector activates less pixels, because the sensors cover a larger area.

In figure 8c & 8d the cluster size of the FE-I4 in x- and y-direction are compared. Here the cluster size in x is with around 1,614 significantly smaller then the size in y with 2,096 pixels activated per hit. This can also be explained by the same reasons as for the DUT & MIMOSA planes. The pixels in y direction are 5 times larger then in x.

So after this cluster investigation one could mention that the expected ToT value for a average cluster of size 2 would be around 14. If one assumes now two tracks going in the same hit cluster an approximated ToT value of 28 would be expected.



(a) total clustersize DUT plane

(b) total clustersize MIMOSA plane 5

(c) clustersize DUT plane in x-direction

(d) clustersize DUT plane in y-direction

Figure 8: Studies of cluster size

- **Cluster Selection:** This step is not included in the analysis of this report, but in general it is possible to exclude clusters from the further process. Reasons for that might be that noisy pixels are included in the cluster which fire at a very high frequency or for example a bad signal vs noise ratio.

3.4 Hit Maker

In this analysis step the expected hits are calculated. An expected hit is just the geometric middle of the cluster as sketched in figure 9.

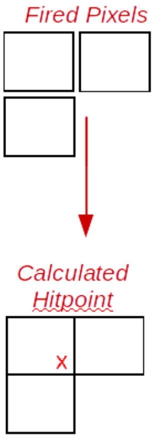
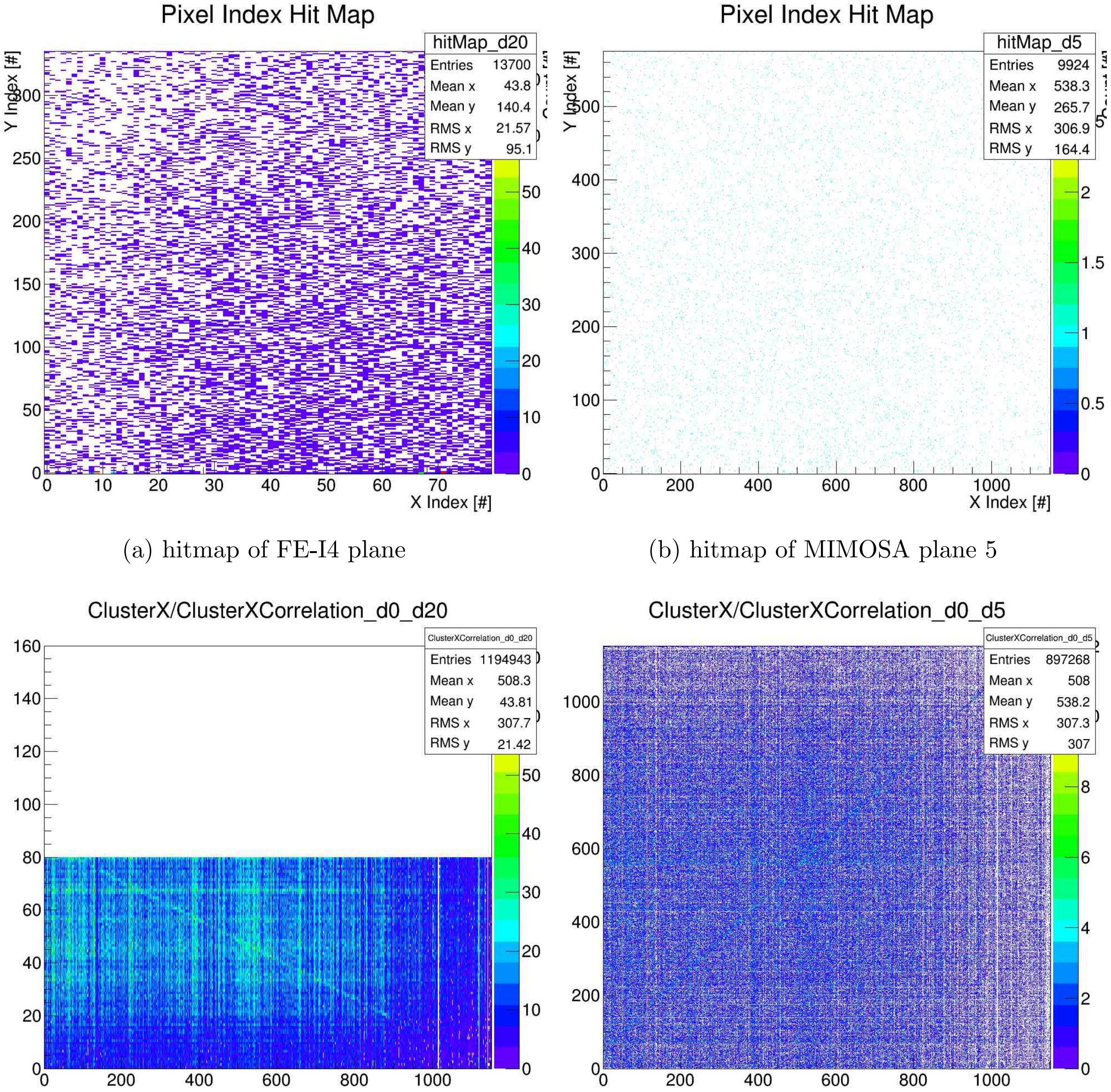


Figure 9: Sketch of the geometric calculated hit position of a track

A other step which is done here is that histograms with correlations between the different planes are created. In figure 10b & 10a the hit maps of the MIMOSA plane 6 and the ATLAS FE-I4 plane are shown. Here is the calculated hit position in x- and y- direction for every cluster of the entire run drawn. There are more hits on the FE-I4 plane which might be a hint of noisy pixels for example. Also the readout time is faster and the death time shorter, which makes it possible that 2 hits in time can be resolved on the DUT where the MIMOSA planes just could measure 1 hit.
Furthermore in figure 10c the correlations between plane 0 and plane 20 (the ATLAS FE-I4) in the x coordinate is sketched. That means every hit point coordinate in x on plane 0 is drawn against all possible x hit-positions on plane 20. For a real track we would expected a correlation between this to positions, if a track hits plane 0 in a certain value and the track is going straight through the telescope without any multiple scattering the value should keep the same on the next plane. So expected correlations are then a line at 45 degrees, but in this histogram a anti correlation is shown, because the FE-I4 plane was tilted around 180 degrees. In the histogram 10d there are correlations shown for two MIMOSA planes as expected.



(a) hitmap of FE-I4 plane

(b) hitmap of MIMOSA plane 5

(c) correlations between MIMOSA and FE-I4 plane in x-direction

(d) correlations between MIMOSA planes 0 & 5 in x-direction

Figure 10: Studies of hit correlations

3.5 Alignment

This processor provides the alignment of the set-up. Because it is not possible to fix the planes of the telescope manually by hand on to a precision in the micrometer regime it gets necessary to align the planes on a software based way. Therefore the a analysis is executed. A couple of tracks out of the hit point collection are build randomly and

the procedure to find the lowest χ^2 distribution (fig.11)between the measured hits and fitted hits is searched. This gives then the alignment geometry of the telescope, so that the tracks can now be reconstructed.
The expected value for the χ^2 over degrees of freedom distribution should be around 1, but the real value of 0,9135 differs from that. A possible explanation for that can be that the beam energy which is also a parameter of the executed analysis was higher then the assumed 12 TeV. A second reason might be that the resolution of the MIMOSA 26 planes in this case was better then the 4 μm which were a parameter of the calculation. They could be also around 3,7 μm for example which would provide a higher error-range.

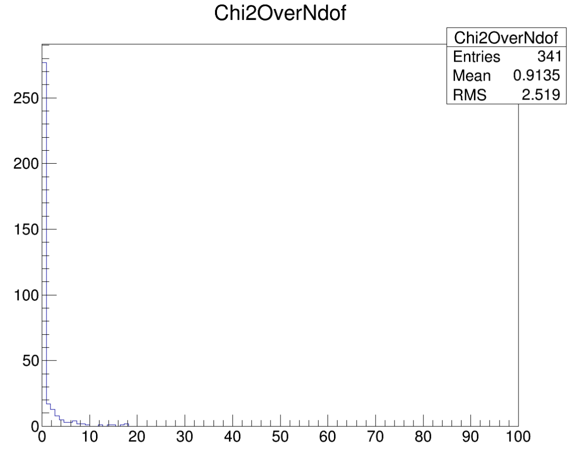
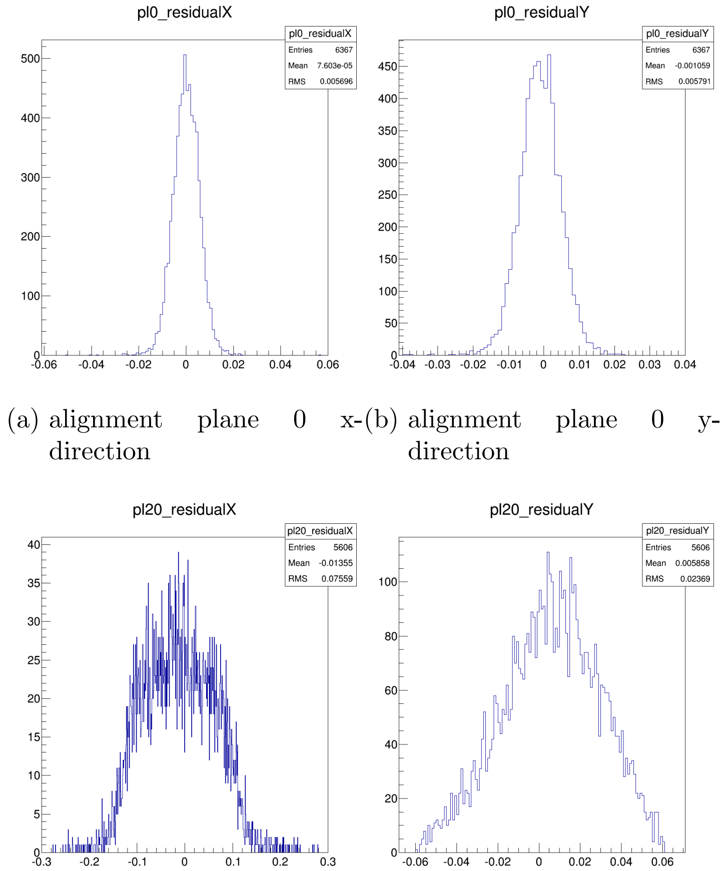


Figure 11: χ^2 over dof distribution

In 12 the residuals of the aligned planes are shown. In the alignment plots of plane 0 which is also taken as reference plane for the alignment of the other planes one can see that they are well aligned with a mean value less then a micron in x/y and with a RMS around 5-6 microns which gives a quality of the resolution of the sensors already.
For the ATLAS FE-I4 plane we have a consistent alignment as well. Here in x-direction the plane is shifted about 14 μm and in y about 6 μm to the reconstructed track. The RMS value is also higher what makes sense, because the pixels are much larger and the RMS is in x with 76 μm and 24 μm in y. That means one has here a wider distribution, but this is still in the agreement with the prediction.



(a) alignment plane 0 x-direction

(b) alignment plane 0 y-direction

(c) alignment plane 20 x-direction

(d) alignment plane 20 y-direction

Figure 12: Studies of plane alignment

3.6 Track Fitter

Here the fitted tracks are created out of the aligned hit collection where now the hits are stored with their positions and the alignment informations. The procedure how the tracks are fitted is the following:

For every hit point on the first plane a cone to the next plane with a special radius is created and in this cone for other track hits is searched and the track with the lowest residual, the smallest value between the measured and fitted hits is chosen as a track candidate. If in this cone no hit point is found then the track might not be reconstructed. It is possible to set the number of missing track hit points for the track reconstructing process as a parameter in the calculation. Also every hit point is just one time usable for fitting and reconstructing the tracks.

3.7 Final Analysis - nTuple

In this last processor the final analysis can be executed. The most important information of the MIMOSA 26 sensors and of the DUT are saved here in root trees. This can for example be the measured hits, the DUT ToT values or the residuals between fitted and measured hit points. In this summer student project also a measurement of the smallest distance between two tracks and a counter for the number of neighbouring hits in a certain area around the track have been added and saved in a tree. With short root scripts histograms can be created out of these trees without changing the source code and recompiling every time.

4 Results

In this section the results of this project will be given. First the finder-radius as a parameter of the analysis will be introduced and then further discussed. Then a second section with the number of neighbours gives first a definition of a neighbour and then their ToT values need to be discussed.

4.1 Finder-radius

In the track reconstruction step a track is reconstructed via a cone going from every plane hit point to the next plane and searches there for hit points as a visualising sketch. In reality all hit points from the 7 planes are projected onto the first plane and for a starting track candidate (a hit on plane 0) all hit points around a certain radius of this hit will be included in the reconstructing analysis and this radius is called finder-radius. Is this radius to small then for many possible hits on plane 0 no hits are found on the other planes and this track will not be found. Is the finder-radius to large then the tracks can not be found because of so called pattern-recognition that means there are to many possible tracks which are used in the analysis and do not find the right track and can not be used in later reconstructing processes so that the other tracks can not be build. This is shown in figure 13 where the best radius for reconstructing most tracks is found at 70 μm. All values lower then 18,4 μm or even 36,8 μm can be called non-physically, because the pixel pitch or the cluster size is then larger as the radius. For the optimal value out of 7645 clusters and 6961 measured tracks 6435 can be fitted, which is equivalent to a efficiency of 90%.

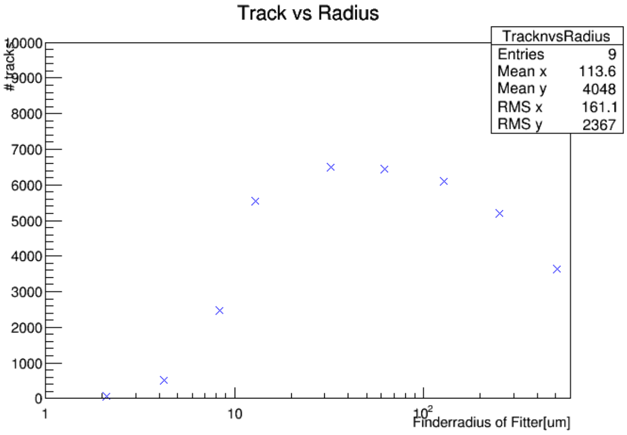
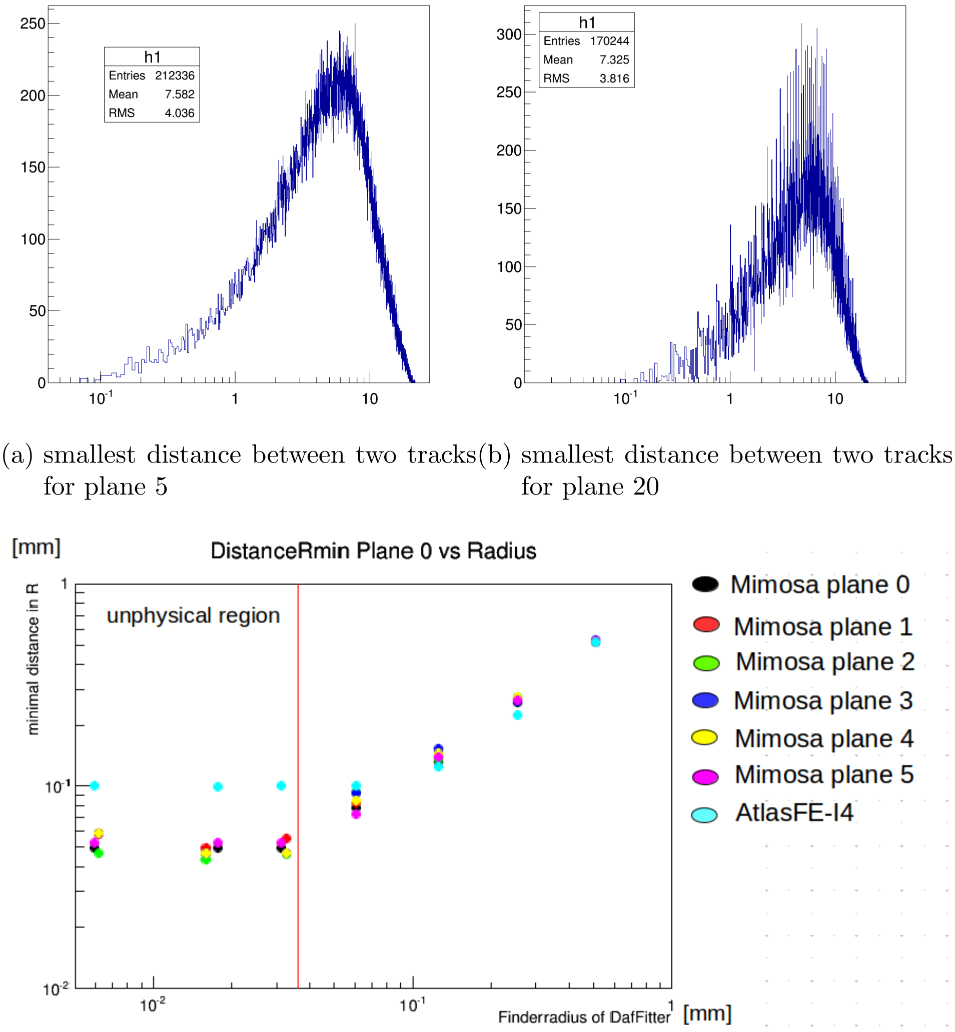


Figure 13: number of reconstructed tracks as a function of finder-radius

In figure 14a the smallest distance between two tracks is shown. For that purpose the distance between every track with all other tracks is callculated and added to the histogram. This gives for the MIMOSA plane 5 a value which seams to be limited by the finder-radius between 70-100 μm. For the ATLAS FE-I4 (14b)the same is done with the result that the smallest distance between two still resolvable tracks is around 100 μm which in this case is more likely limited by the pixel sensor. These have in y-direction a pitch of 50 μm and for having two separate tracks one pixel in between has to be not activated, because otherwise they would build a cluster and could without cluster splitting give only one track. This can be seen in plot 14c where the minimal distance does not change between two tracks for a finder-radius below 100 μm for the FE-I4 and for the MIMOSA sensor below a radius of 36,8 μm. Above the finder-radius of 70 μm the minimal distance increases again because of pattern-recognition effects and take approximately the same values as the finder-radius which gives there a linear dependence. Here the predictions and the measurement agree quite good.



(a) smallest distance between two tracks for plane 5

(b) smallest distance between two tracks for plane 20

(c) smallest distance as function of finder-radius

Figure 14: Studies of finder-radius

4.2 Neighbor studies of a track

In this section the number of neighbours of a track will be analysed and their properties will be discussed. A neighbour of a track is defined as a measured hit point which is adjacent to the track, but does not belong to the track, that means this hit must be at least around 2 pitches away from the track, because otherwise it would belong to the same cluster and with that to the track. To find such a neighbour of a track for every hit point in a plane the algorithm searches for measured hits in the defined area in the hit collection and all neighbouring hits off all MIMOSA planes are counted. The area is defined trough a minimal distance in x- and y- direction around the track hit separately. For the first run the minimal distances are in both directions 36 μm. In figure 15 the number of tracks with a certain amount of neighbours is plotted. In total there are 1457 tracks with neighbours out of a track collection of about 210.000 tracks which means that less then 1% of the tracks have neighbours in a 2 pixel pitch.

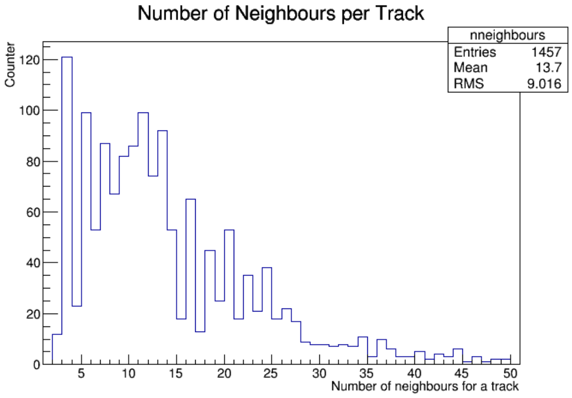
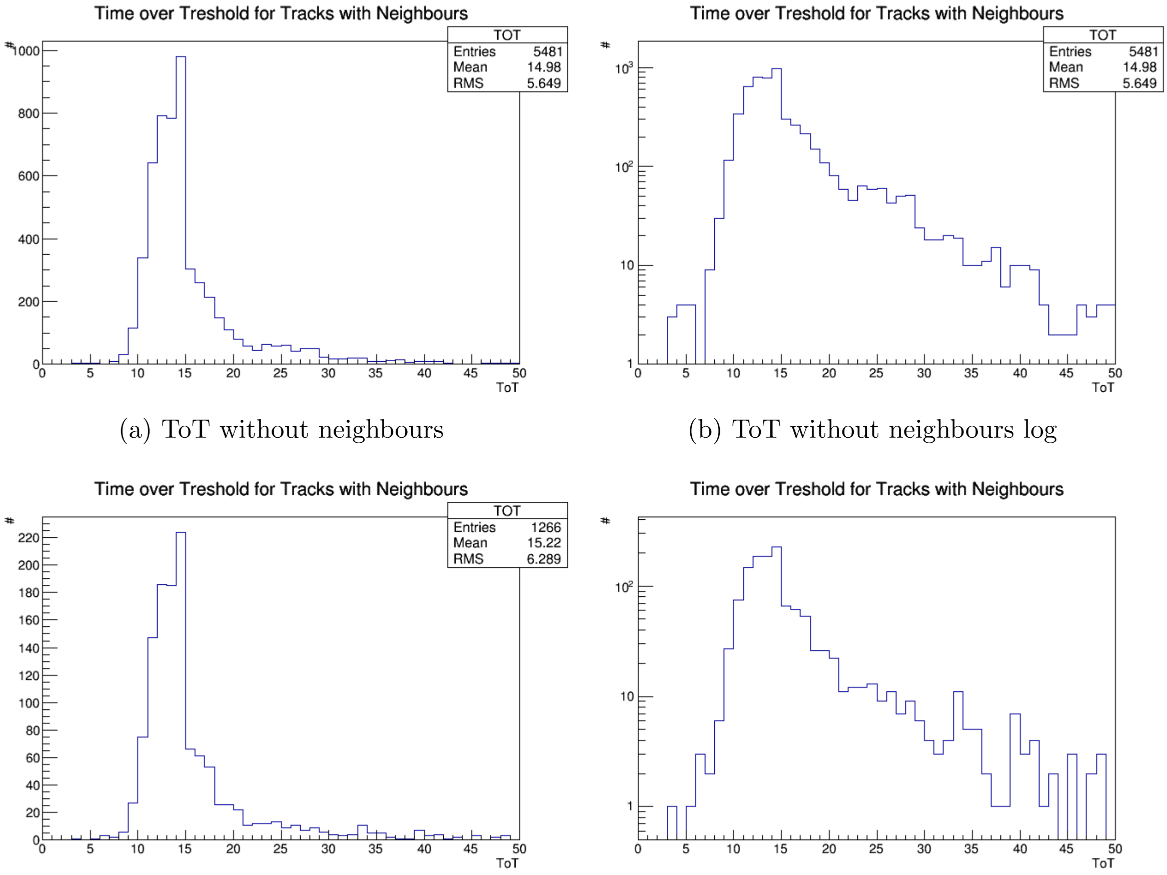


Figure 15: number of tracks with neighbours

In the graphs of figure 16 the Time over Threshold values of the tracks without any neighbouring hits 16a & 16b are compared to the tracks which have some hits around itself (16c & 16d). This ToT value is measured by the ATLAS FE-I4 sensors. For every track with a certain amount of neighbours one wants to study the ToT is read out and therefore the plane of the DUT must be found. This information is stored in the track collection and has to be read out.

As mentioned earlier in the report the main idea was to find clusters with two tracks going into them and study their ToTs. If the average ToT of a single track without any neighbours is assumed to be 14 then a second peak for tracks with 6 or more neighbours which build the adjacent track, is expected at a ToT number of 28. This peak then should be stronger in the collection of tracks which have neighbours, but this can not be verified by these graphs. To low statistics could be one explanation here, but a single study of tracks with exactly 6 adjacent hits could also be interesting and give new hints.



(a) ToT without neighbours

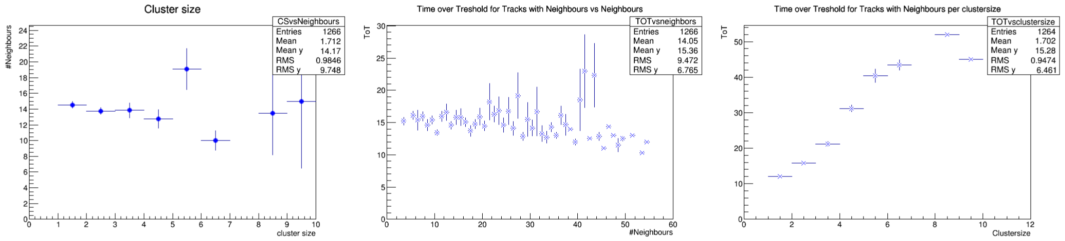
(b) ToT without neighbours log

(c) ToT with neighbours

(d) ToT with neighbours log

Figure 16: Time over Threshold without & with neighbours

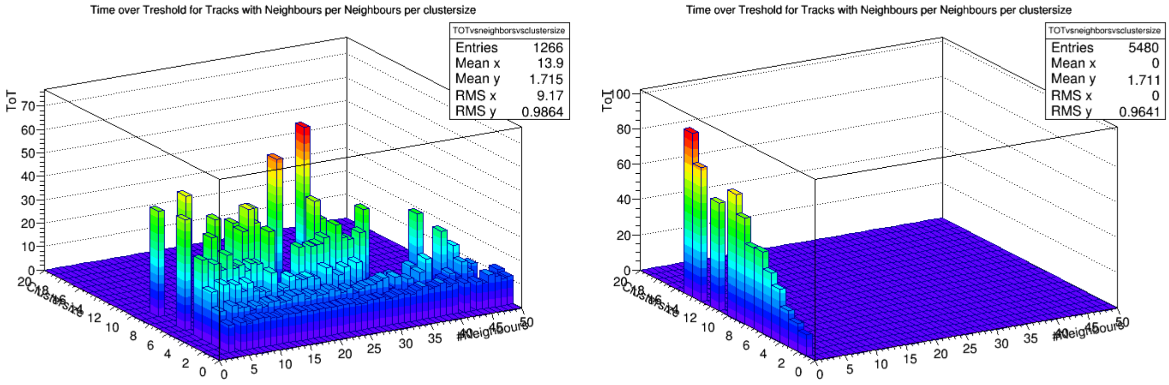
In the next plots 17 for example the correlations between cluster size and neighbours are given in 17a. Here the cluster size has no influence of the number of neighbours or vice versa which is not obvious, because the chance of having more neighbours around a bigger cluster is given. That there is no correlation might linked again to to low statistics and noise (tracks with less then 6 neighbours). An other explanation might be that for more neighbours in a bigger cluster also more neighbouring tracks still are needed which occur only on very low rates.
In graph 17b the ToT is drawn against the number of neighbours of a track and also here there is no real dependence between those two variables. It is more likely that for a higher number of neighbours higher then 40 the average ToT even gets less. That might come from the fact that the charge is then forwarded to the surrounding other clusters. In the last plot 17c the ToT is shown as a function of the cluster size where this is up to a cluster size of 9 a linear dependence. This is as expected, because the ToT value of every pixel is added up to the value for the hole cluster. As more pixels are hit as more charge is collected and so higher gets the ToT.



(a) neighbours vs cluster size (b) ToT vs neighbours (c) ToT vs cluster size

Figure 17: Time over Threshold with neighbours

The results of the previous graphs in figure 17 are again shown now in 3D graphs in figure 18. Here the ToT is given in dependce of the numbers of neighbours in the x-axis and the cluster size on the y-axis. Again the ToT value is a linear function of the cluster size, but seams to be independent of the number of neighbours.



(a) ToT with neighbours (b) ToT without neighbours

Figure 18: Time over Threshold vs neighbours vs clustersize

5 Conclusions

The analysis of a high occupancy test beam was best executed with a finder-radius of 70 μm and with two parallel beams. Here the track finding reconstruction process has a high efficiency of 90%. The first results do not indicate a distinct raise of ToT with increasing number of neighbouring hits. There is also no correlation so far between the number of neighbouring hits and the cluster size which was expected to be proportional. For these tracks with neighbours there is so fare only very low statistics available which should be taken into account.

The possibility is given for further studies here, the test beam data taken at PS T9 during the summer student program with the same set-up as at SLAC still can be analysed and should provide higher statistics. Furthermore the number of neighbouring hits could be included to the track reconstruction analysis as a parameter which allows to split clusters.

Acknowledgements

I would like to thank my Supervisor Igor Rubinskiy, who had to guide me through a lot of questions coming up during my project during the summer school here at DESY and making this project interesting and especially instructive. Also I want to thank the DESY telescope group here at DESY for joining the weekly summer student presentations and giving there many good hints how to continue.

I want to thank Phillip Hamnett for his support during the testbeam at PS, Alex Morton and Richard Peschke for their help in C++ and Eda Yildrim for many tutorials.

Furthermore I would like to thank the summer students Simon de Ridder and Michael Nelson which shared their office with me and additional to the nice working atmosphere helped me at some point.

I want to thank the hole ATLAS group for their warm welcome and including us to their social activities making us feel like a part of their community.

Last but never the least I want to thank the organizers of the summer school program which made this educational and nice stay here in Hamburg possible and the lecturers which showed a lot of interesting research fields in particle physics.

References

[1] Julia Rieger *ToT Calibration for IBL,* Georg-August-Universitt Goettingen, 07.07.2014

[2] http://eutelescope.web.cern.ch

[3] Eda Yildrim https://indico.cern.ch/event/257146/material/slides/0.pdf

[4] Paul Schuetze www.desy.de/f/students/2013/reports/schuezte.pdf

[5] Tobias Bisanz www.desy.de/f/students/2013/reports/bisanz.pdf