# Reconstruction and Analysis of data taken from a Large Prototype TPC for the ILC

**Summerstudent Program 2009, DESY**

Tessa Charles[a]

[a] Monash University

September 11, 2009

**Abstract**

This report describes how with the use of a steering file, data taken from the Large Prototype (LP) Time Projection Chamber (TPC) at DESY was reconstructed for analysis using various MarlinTPC processors. The goal of this work was to verify the correct operation of various processors by using real data taken with the DESY 5GeV electron test beam. TPCPulses and TPCHits of the LCIO classes `TrackerPulse` and `TrackerHits`, were successfully reconstructed with insight into rudimentary Track reconstruction also described in this report.

## Contents

# 1 Introduction

This work was done as at DESY, Hamburg within the FLC/TPC group. This group focuses on Research and Design (R&D) of the Time Projection Chamber (TPC) as a component of the International Large Detector (ILD) for the proposed International Linear Collider (ILC). This report focuses on the testing and verification of various processors of the software framework MarlinTPC.

This report also describes some of the basic operating principles of the TPC as well as describing the process needed for obtaining information regarding the Hits, Pulses and Tracks recorded by the LP detector.

## 1.1 Time Projection Chamber

The Time Projection Chamber (TPC) is the central tracker of the ILD design and allows three dimensional tracking of the trajectories of charged particles. It is comprised of a large gas filled cylinder with an anode and cathode at opposing ends. Particles that enter the chamber ionise the gas, which is usually around 90% noble gas and 10% quencher, releasing electrons that drift under the influence of the high magnetic field, toward the anode where they are detected [1].

In order to achieve a homogeneous field distribution for accelerating the drift electrons, a field cage is used. The field cage is comprised of conducting rings around the circumference of the cylinder, which gradually step down the potential from the cathode to the anode (figure 1).

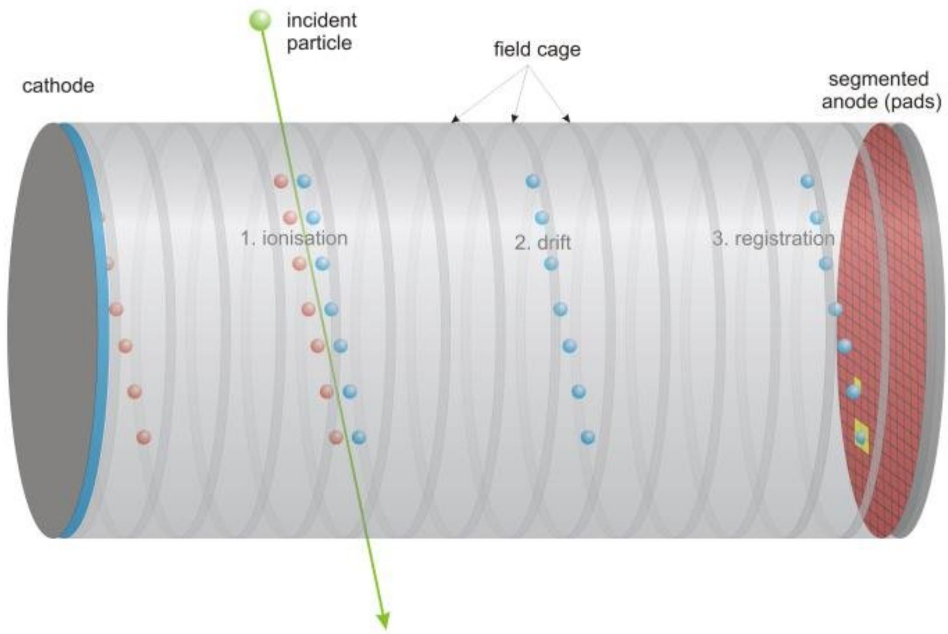


Figure 1: *Working principle of a TPC*

A large magnetic field is used to bend the trajectories of the particles. By measuring the curvature of the trajectories and with knowledge of the field strength, the momentum of the particle can be easily calculated.

The anode consist of segmented pads where Gas Electron Multipliers (GEM) are used to amplify the signal. Upon arriving at the detector, the drift electrons are in the order of 100s of electrons per square centimeter, therefore an effective amplification device is needed. GEMs structure is a thin Kapton foil coated with a layer of copper on both sides and perforated with holes typically of 70 $\mu$m diameter with a pitch of 140$\mu$m. A voltage is applied to the cooper layers, usually of a few 100 volts, resulting in an electric field of around 10 kV/cm. This field accelerates the electrons causing secondary ionisation, amplifying the signal.

## 1.2 Software/Computing

Until recently there has been a vast array of simulation, reconstruction and analysis software available for use within the ILC TPC scientific community. The different data formats produced by the alternative systems had the direct result of time consuming and error prone work in converting between data formats or computing codes. As a result a universal software framework MarlinTPC, is being developed to allow the easy transfer of information and data between the relevant groups [1].

### 1.2.1 LCIO

In order to standardise the exchange of data amongst different software frameworks, the LCIO (Linear Collider I/O) persistency framework was created. This works to transfer the data in its most basic form (in memory objects) to a permanent database. C++ and Java implementation can occur within a common interface (API). A Fortran interface to the C++ implementation is also available. The LCIO allows for a highly modular system to be created, allowing the user to limit the number of processors needed for reconstruction and analysis by only selecting the processors essential for the particular task [2].

### 1.2.2 GEAR and LCCD

Gear (Geometry API for Reconstruction) is a data storage toolkit used for ILC reconstruction software. The information stored in GEAR is static information such as the TPC dimensions and geometry [3]. Information that is subject to change such as drift velocities, magnetic field and metrological data, is stored in the LCCD (Linear Collider Conditions Data) toolkit [4].

## 1.3 Marlin

MarlinTPC is a highly modular software framework for simulation, digitization, reconstruction and analysis. This allows for the individual modules (or processors), to be developed by different people and the data produced and algorithms can be easily compared despite the large diversity of readout structures, electronics and amplification systems used between the various ILC TPC groups [5]. The different processors analyse data objects in a collection named LCEvent and in turn create an output collection which is added to the event.

## 2 Test Beam Data

The data analysed in this report was taken in April 2009 using the GEM structure of the University of Saga and KEK in Japan, and using the DESY electron test beam. Four separate runs, each consisting of 20000 events, were used to analyse the Marlin processors. Consistent to all four runs was that the data taken was zero-suppressed with a 5 GeV beam and T2K gas was used in the TPC. Further parameters are listed in table 1.

| **Run number** | **Magnetic field (T)** | **Drift length (mm)** |
|---|---|---|
| 6850 | 1 | 150 |
| 6864 | 0 | 150 |
| 7050 | 1 | 100 |
| 7065 | 1 | 400 |

Table 1: Parameters that differed between the runs lists.

## 3 Data Reconstruction

In order to activate the desired processors in the correct order, a steering file was needed. Within the steering file, the processors needed are listed in order of application, as well as the global parameters and more specific local parameters and inputs required for individual processors.

#### 3.0.1 Processors

Every processor performs a unique task. As such, many processors are often needed to obtaining meaningful information from the inputed data. As mentioned above, the processor uses data from LCEvent, creating additional output collections as seen under Collections Name in table 2. These modules allow the user to call on one processor at a time using a steering file. The processors in table 2 are written in order of application. That is, a steering file would most likely begin with the processor `TrackerRawDataToDataConverter`, which acts on the `TrackerRawData` objects which are stored in TPCRawData. The `TrackerRawDataToDataConverter` then produces a collection of `TrackerData` objects in the collection TPCData.

The first processor in the reconstruction is the `TrackerRawDataToDataConverterProcessor`. The role of this processor is to convert the ADC counts into `TrackerData` objects with floating point numbers. At this point non zero-suppressed data require a Pedestal subtraction processor using pedestal information read in by the ConditionsProcessor.

| Data Structure | Processor Name | Collection Name |
|---|---|---|
| TrackerRawData | TrackerRawDataToDataConverter | AltroRawData |
| TrackerData | ADCPulseConverter, ChannelMapper | TPCData |
| TrackerPulse | HitTrackFinderTopoFinder | TPCPusles |
| TrackerHit | SimpleTrackSeeder | TPCHits, TPCTrackCandidates |

Table 2: MarlinTPC Processors used in data reconstruction.

The next logical step is to find the Pulses using the `ADCPulseConverter`. The Pulses are defined as the pads where a signal has been detected above a certain threshold. This processor calculates charge and time information pertaining to the Pulse. Another important processor is the `ChannelMappingProcessor`which converts the hardware channel numbers of the electronics to logical/software pad indices with the aid the of the GEAR file.

The `HitTrackFinderTopoProcessor` performs a topological search for Pulses and clusters them into Hits. Pulses on contiguous pads are grouped together along a row of pads as Hit candidates. The coordinates of the Hit are then calculated with reasonable confidence that the Pulses belong to the same event incident. The minimum number of Pulses to make up Hit was set to one. This ensures that there exist an anomalous track traveling perpendicular to the rows, the `TrackFinder processor`(which acts on the Hits found) would still be capable of recognising the tack.

The `SimpleTrackSeederProcessor` estimates the track parameters forming a rudimentary basis for the Track fitting to follow. Finally the `TrackFitterLikihoodProcessor` determines the track parameters by maximising the global likelihood for observing the measured charge distribution on all pads associated with the track [6].

## 4 Results

The test beam data was successfully reconstructed to produce TPCData, TPCPulses and TP-CHits collections. The Pulses and Hits are shown in figures 2 and 3 respectively. Similar images were obtained for all four runs. The colour variation across the Pulses in figure 2, correspond the variation in ADC counts deposited on the pads. More specifically, one ADC count corresponds to a charge deposit of 83 fC. This information could then be used with known conditions data to calculate the dE/dx value in future, yet to be developed, processors.

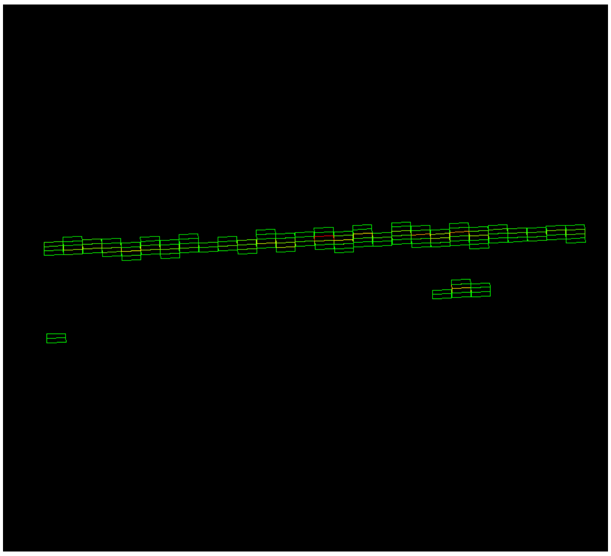

Figure 2: *Pulses shown for one detected event.*

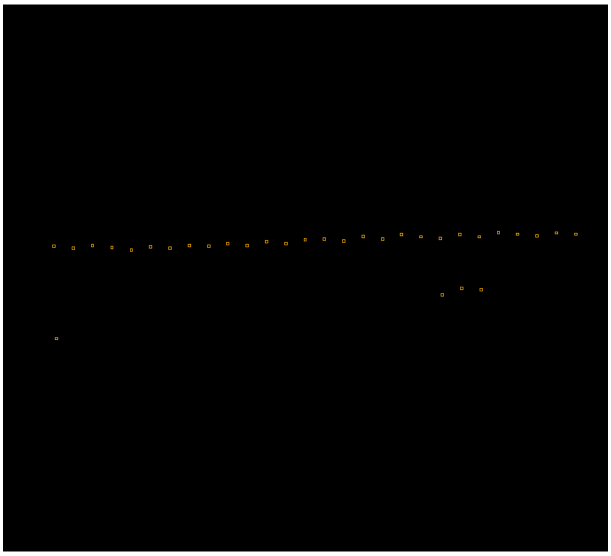

Figure 3: *Hits shown for one detected event*

Figure 3 also shows a slight curvature in the track, bending upwards at one end of the module and down at the other. This was most likely due to the fact that the gating GEM of the double GEM structure, was not used creating a perturbation in the electric field lines near the anode surface. As a result the electrons entering the GEM near the edge of the module were shifted causing this alteration to the electrons' track.

Many events were also able to be displayed simultaneously, as shown in figure 5. This illustrates well the hit positions centering around the weighted median of the Pulse charge distribution.

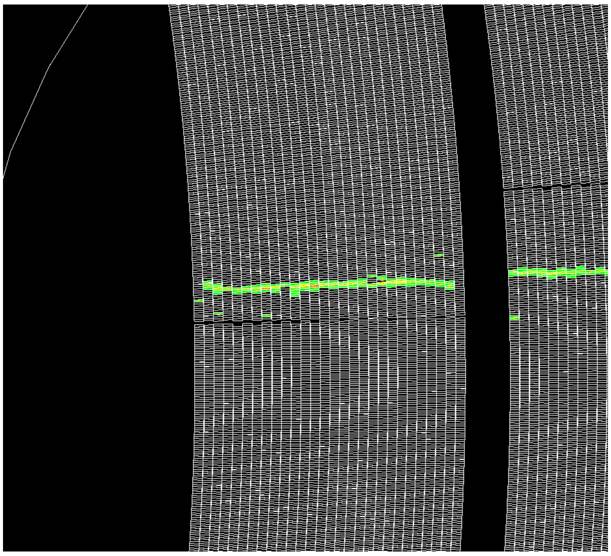

Figure 4: *Event showing curvature near module edges.*

The same reconstructed data was represented in histogram form in order to more effectively compare the four data runs. These are shown in figures 6 through to 9. By close inspection of the x distribution histograms, it is possible to see the 3 modules of the end plate which is consistently measured to be approximately 150 mm for all four runs. Interestingly run number 7050 and 7065, both recorded information at a radius of approximately 1600 mm which should correspond to the gap between modules 3 and 6. This is not yet fully understood and further work is needed to determine the reason behind this seemingly nonsensical result. It is possible that the GEAR file's pad geometry is not exact, causing the histogram data to appear as if modules 3 and 6 overlap.

The width of the y distribution histograms, indicates the variation in the track position of the numerous tracks that entered the TPC. On average this various is approximately 80mm. This indicates that the electron test beam, which was not subjected to any focusing devices upon leaving the pre-accelerator DESY, diverges significantly.

One unresolved queries remains in attempting to understand the physical meaning of the z distribution scale. If it is assumed that the z direction, i.e along the length of the cylinder, is determined by being proportional to the number of time buckets counted between the trigger system being activated and the drift electrons being detected. However a conversion factor is needed in order to obtain the z distances in mm. To complicate this matter, the readout frequency for the data taken is not known and where this information should be used is also unclear. A default value of 40 MHz is found in the GEAR file however the `ADCPulseConvertorProcessor`, which is the only processor that refers to the readout frequency, does not obtain this information

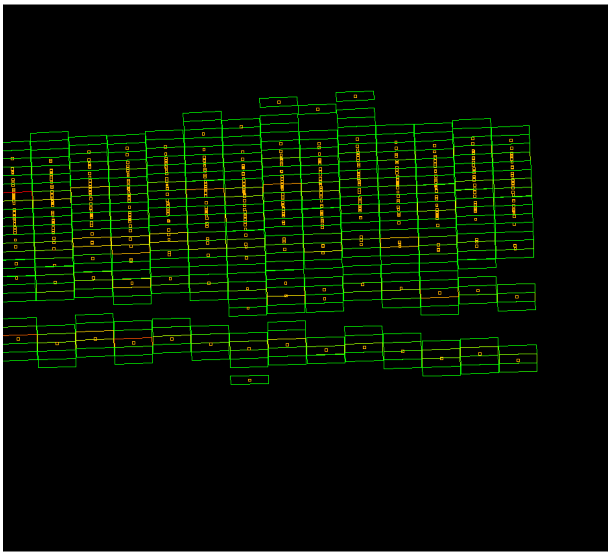

Figure 5: *Many tracks shown detected in run number 6864.*

from the GEAR file, but rather uses the result of simple arithmetic involving the module number (i.e. 1e+9/module). Similar confusion surrounds the drift velocity which also obtains a default value of 4.0e+07 mm/$\mu$s in the GEAR file. The HitTrackFinderProcessor contains an "optional" parameter input *vDrift*, however this is a required input value with no reference made to the drift velocity indicated in the GEAR file.

No significant differences were observed between the run with zero magnetic field and the runs with 1 T applied. This is as expected due to the small length of the track (of only three modules) and the high energy of the test beam, the 1 T field did not have a noticeable effect on the incident electrons.

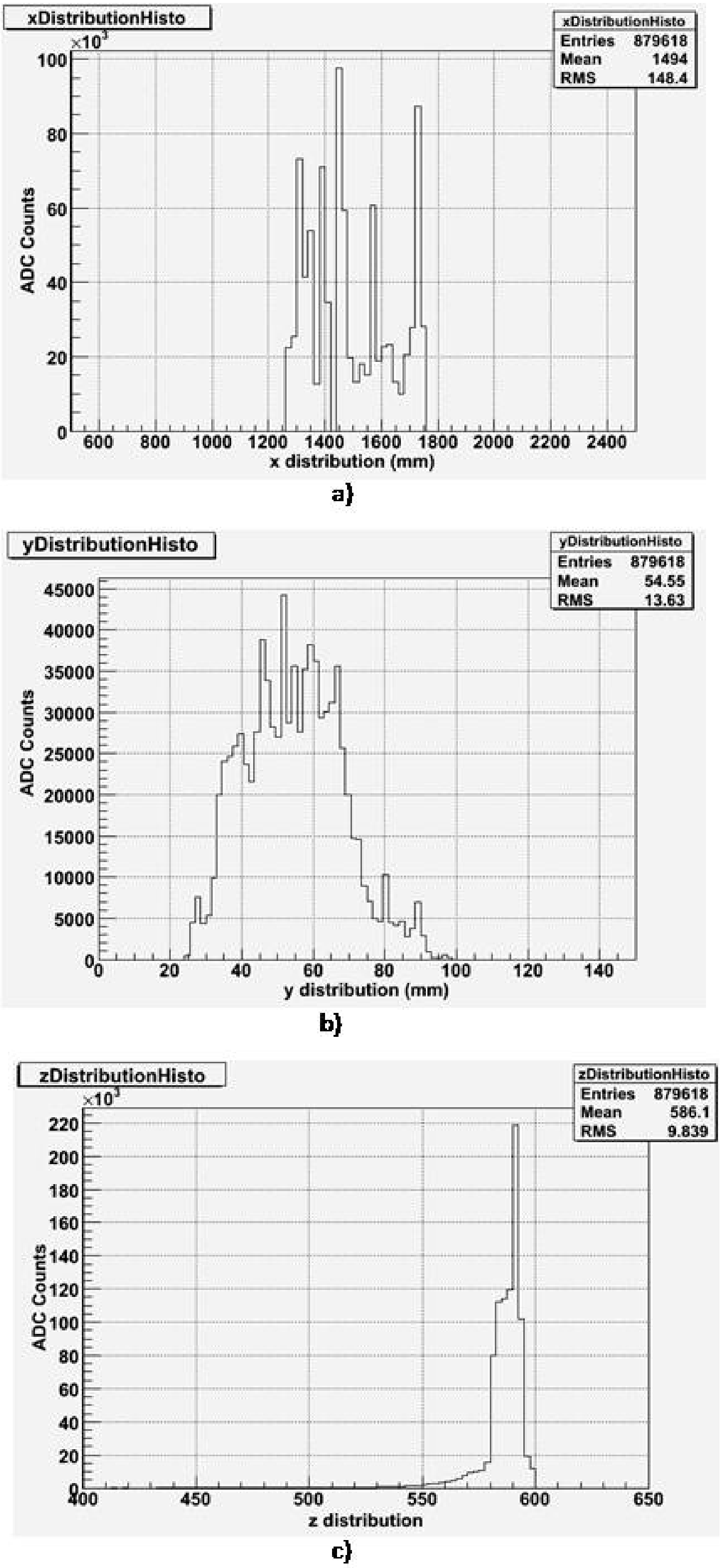


Figure 6: *ADC information of Hits from run number 6850, a) x distribution b) y distribution c) z distribution*

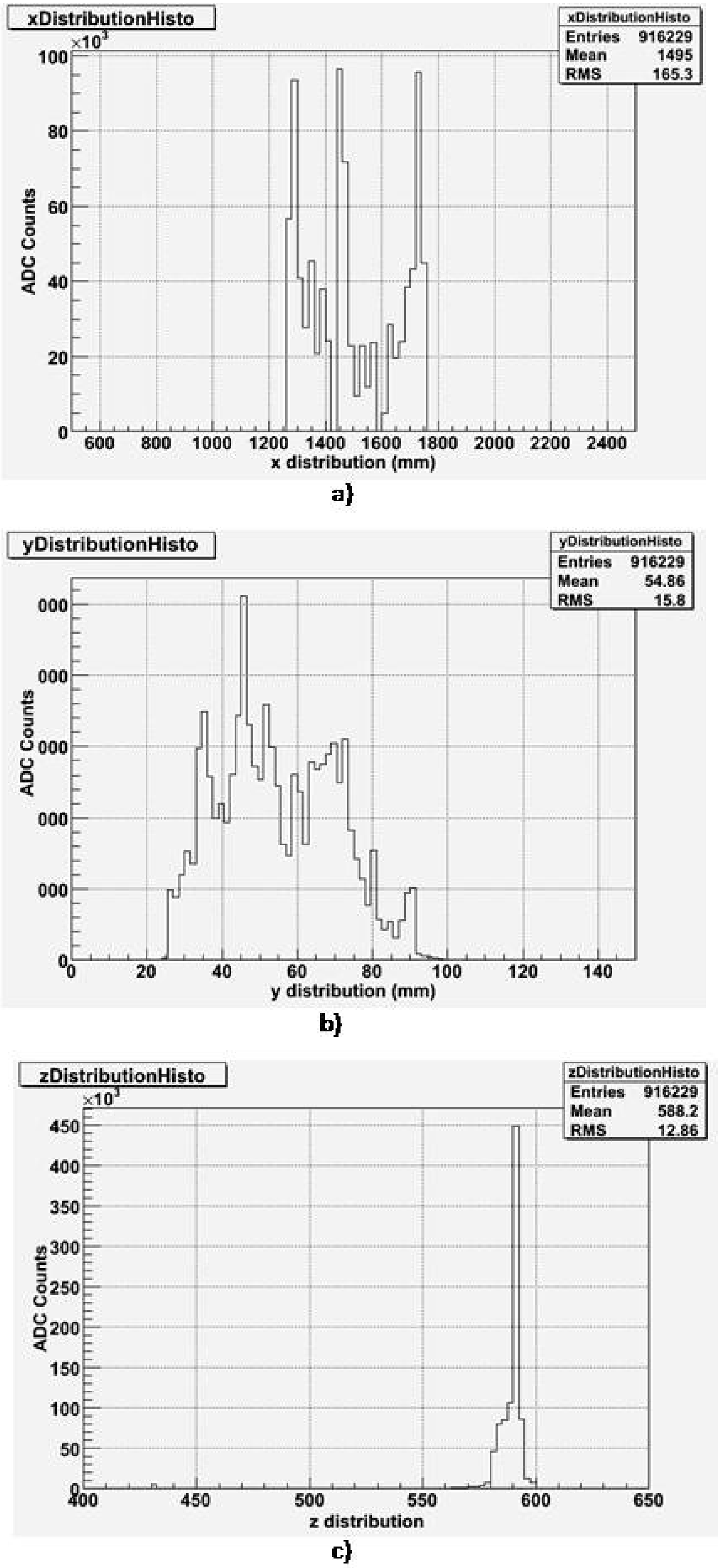


Figure 7: *ADC information of Hits from run number 6864 showing a) x distribution b) y distribution c) z distribution*

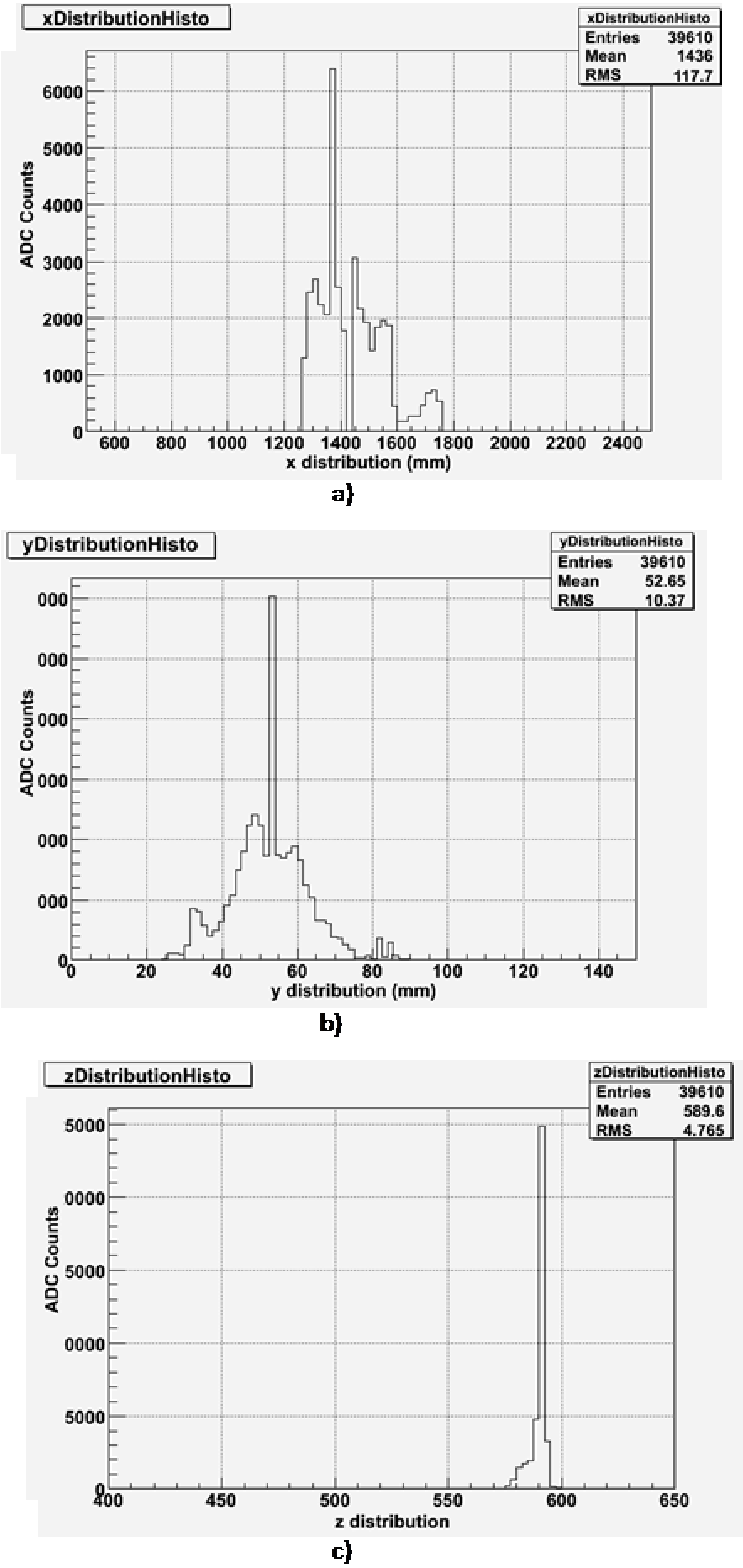


Figure 8: *ADC information of Hits from run number 7050 showing a) x distribution b) y distribution c) z distribution*

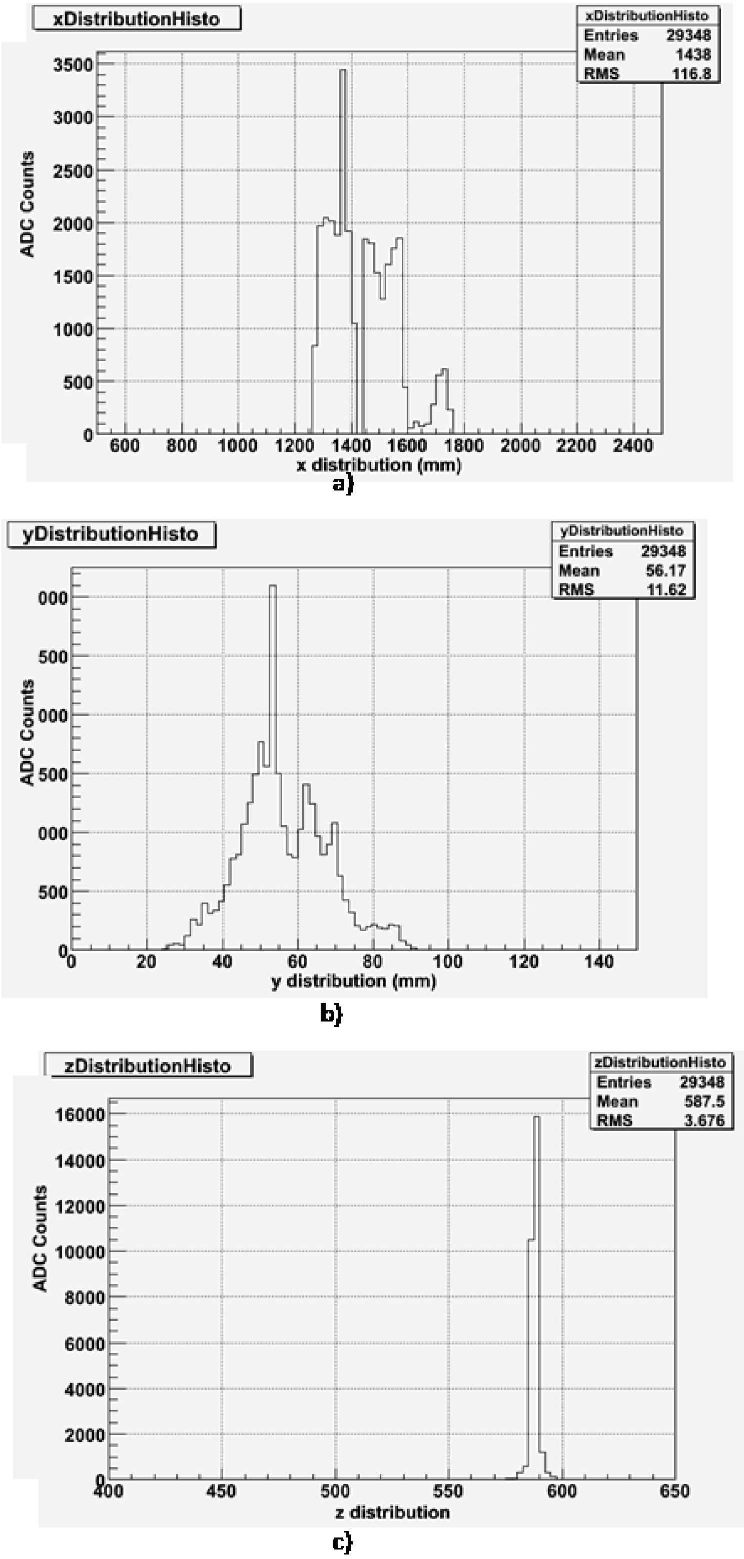


Figure 9: *ADC information of Hits from run number 7065 showing a) x distribution b) y distribution c) z distribution*

Figure 10 shows a *curler* which was detected in event 2 of run number 6850. This occurrence was due to secondary ionisation of the drift electron producing an electron of low momentum, thus being more influenced by the magnetic field and undertaking a high curvature track. This is further verification that the processors successfully reconstructed the data despite the large curvature of the track and the fact the there were two tracks in the one event. This is positive reinforcement that such detail could be detected using the prototype TPC and Marlin software framework with processors.

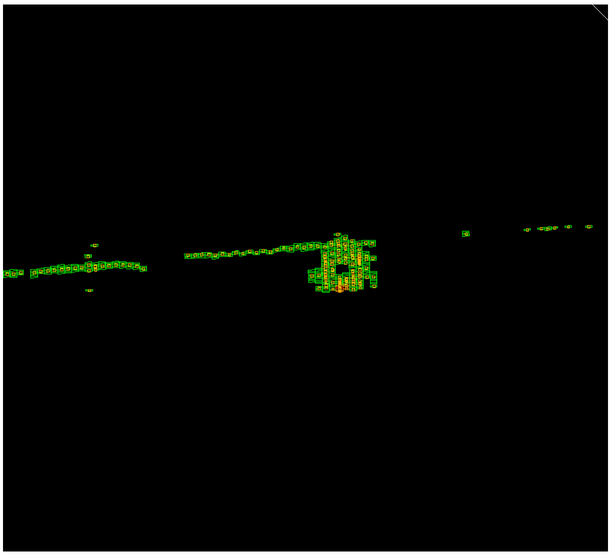

Figure 10: *Curler found in event 2 of run number 6850*

## 5 Conclusion

Data taken from the Large Prototype TPC at DESY, was successfully reconstructed to form the data objects of the collections TPCHits and TPCPulses. This was done by creating a steering file to guide the software framework MarlinTPC to activate various processors. By completing this task the following processors were verified:

- `TrackerRawDataToDataConverterProcessor`
- `ADCPulseConverterProcessor`
- `ChannelMapperProcessor`
- `HitTrackFinderTopoFinderProcessor`
- `ConditionsProcessor`

Further studies are required to determine the physical significance values of the z distriubtion of the Hits recorded. Continuation of this work should see the data collections TPCSeedTracks and TPCTracks produced using further MarlinTPC processors.

# 6 Acknowledgements

I would like to thank my supervisor for this project Klaus Dehmelt for organising this task for me and his much appreciated help and guidance along the way. Also thank you to the FLC group for allowing me to undertake this project at DESY and finally I would also like to extend my thanks to Stewart Martin-Haugh for specific help with LINUX queries and helpful discussions.

# References

[1] FLC-TPC homepage `http://www-flc.desy.de/tpc/`

[2] LCIO homepage `http://lcio.desy.de`

[3] GEAR homepage `http://ilcsoft.desy.de/gear`

[4] LCCD homepage `http://ilcsoft.desy.de/lccd`

[5] MarlinTPC homepage `http://ilcsoft.desy.de/portal/software_packages/marlintpc`

[6] MarlinTPC: A Marlin based common TPC software framework for the LC-TPC collaboration *J.Abernathy et al* EUDET-Report-2007-04