June Test Beam: CERN GDD preliminary data analysis

On behalf of CERN GDD group

Overview

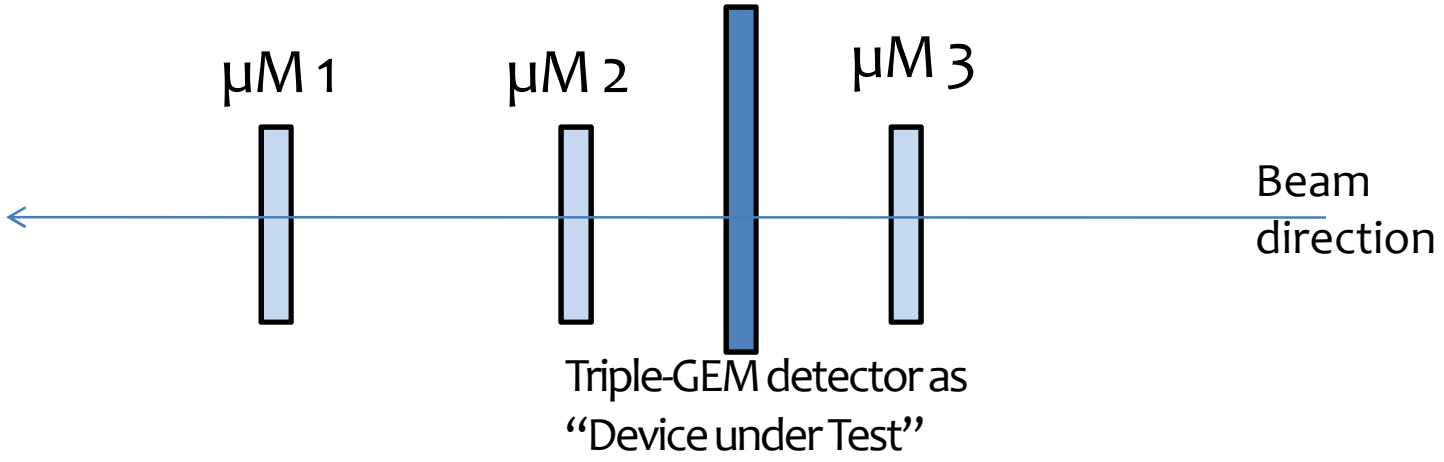
- The CERN GDD goal for June test beam was:
  - to understand the performance of the telescopes provided to the users
  - to start to build an analysis framework

➢ The work has been focused especially on the organization of the algorithms and tools

➢ We have just started to analyse the first datasets

A reminder of the setup

- 3 μM stations to reconstruct the track
- A device under test (two different Triple-GEM detectors) where the track information is used

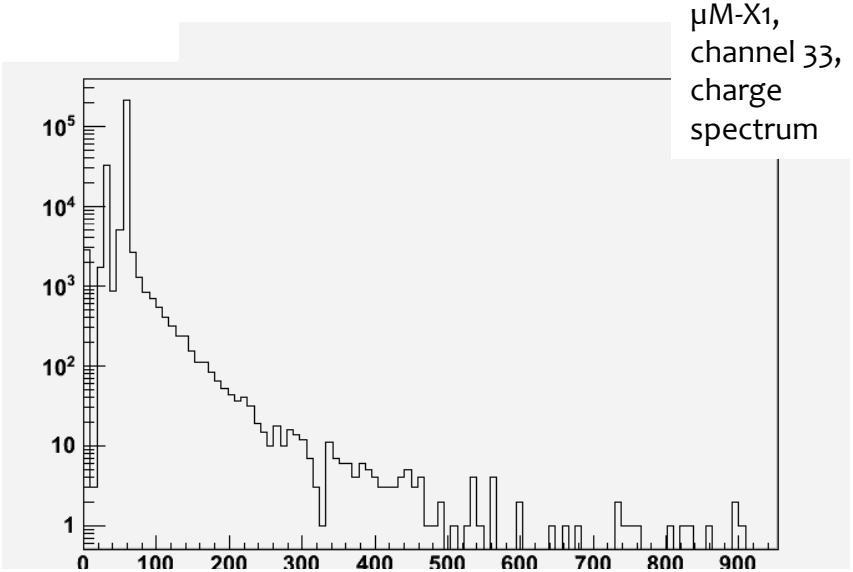


- All detectors were read out with Gassiplex FEE in order to use the same DAQ

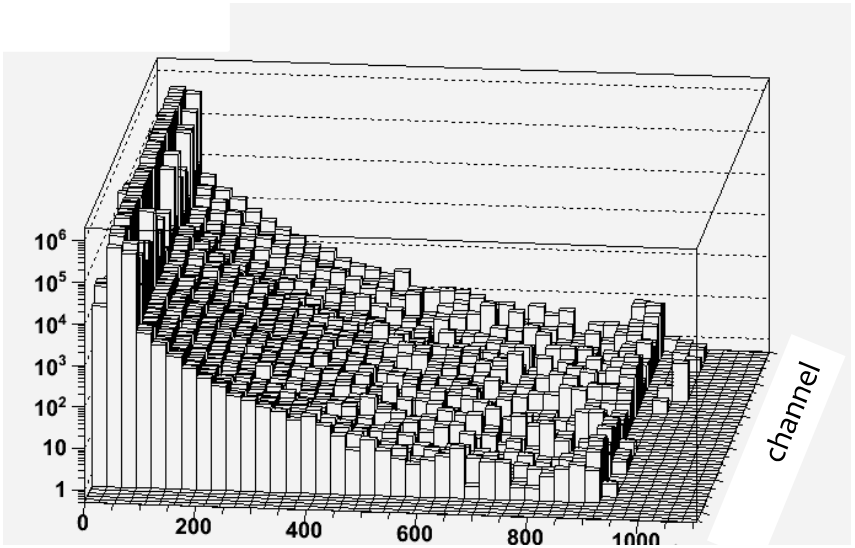
Software implementation

- All the analysis is performed with the ROOT package
- Raw data are converted from the binary DAQ format to a ROOT tree
- Each Gassiplex data are converted in an array of 96 elements, each giving the charge of one channel of the card.

Some examples of raw data..



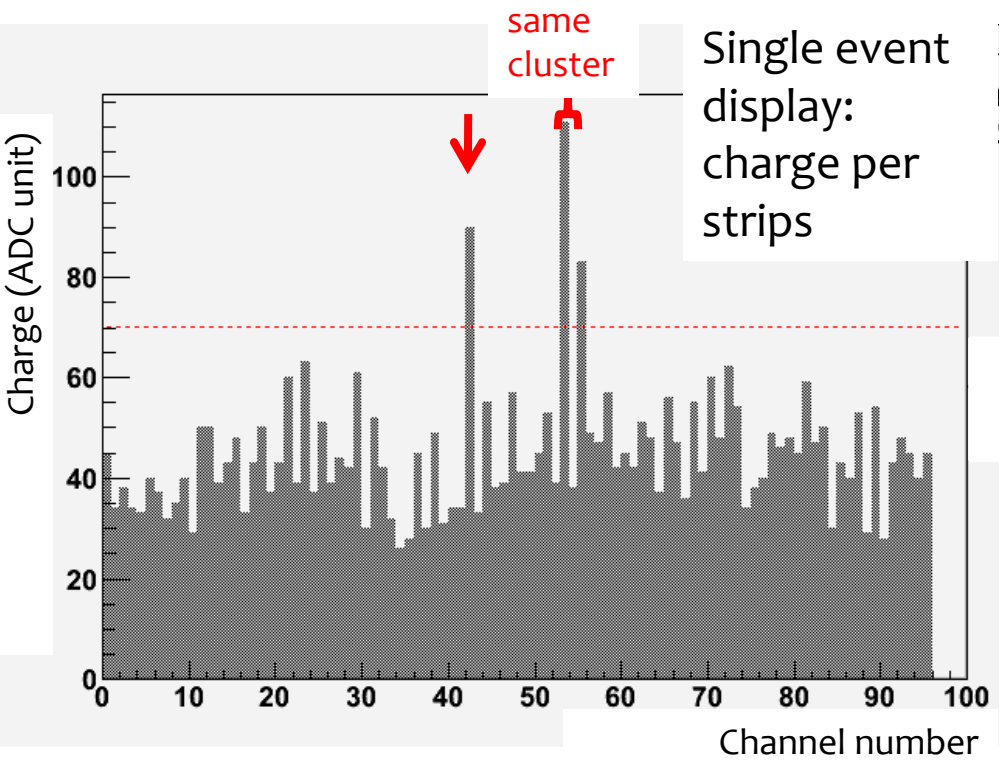
Charge spectrum of 1st x-strips µM, channel 33



Charge spectrum of 1st x-strips µM, all channels

A simple clustering algorithm

- Channels over a fixed threshold create a cluster
- Neighbouring over-threshold channels belong to the same cluster
- Cluster charge is the total charge, cluster position is the charge centre of gravity
- Two free parameters to be optimized:
  - Threshold
  - Max distance between two channels of the same cluster

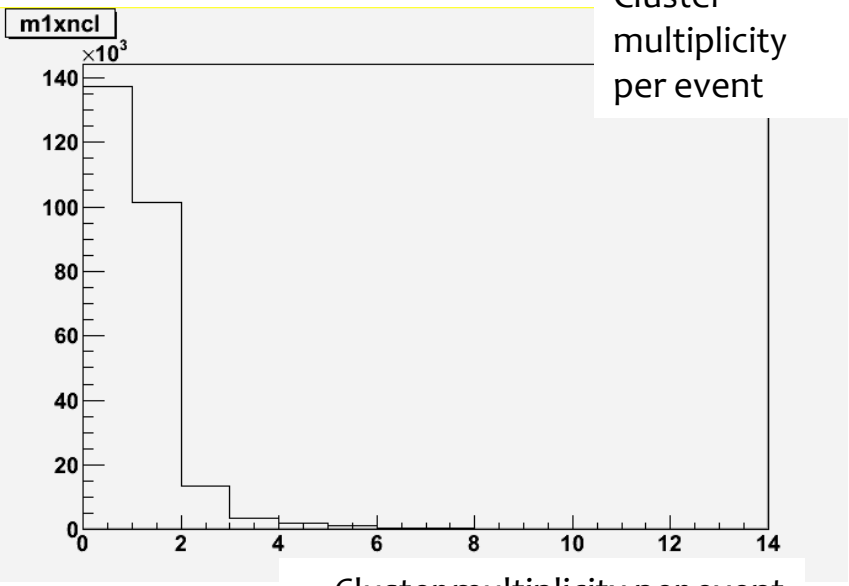


A “Cluster” class

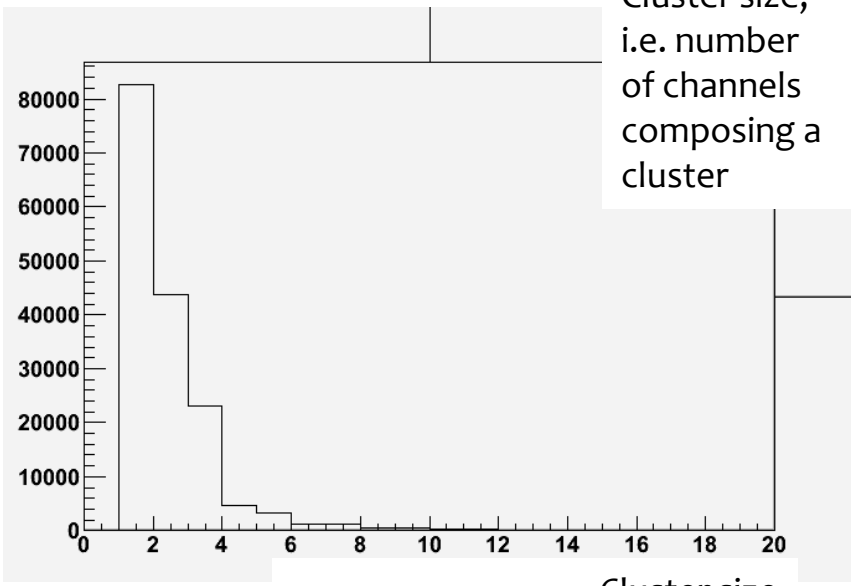
- “Cluster” objects are able to “Find()” themselves in an array of channels
- Easy to implement more sophisticate algorithms, while keeping the same interface
- .. and all the other advantages of object oriented programming

```
class Cluster
  : private TC
{

public:
  short firstch;
  short lastch;
  short maxpos;
  short maxq;
  float pos;
  float q;

  //Default constructor
  Cluster (short firstch = -1, short lastch = -1,
           short maxpos = -1, short maxq = 0,
           float pos = -1., float q = 0.);

  //Copy constructor
  Cluster (const Cluster& original);

  int Find(const short * arrayofch,
           const short startch, const short maxnumofch,
           const int threshold, const int rangecluster,
           short * lastcheckedch = 0);
  static float CentreOfGravity (const short * arrayofch,
                                const short startch,
                                const short endch,
                                float * totalcharge = 0);
```

New information after running clustering algorithm



Cluster multiplicity per event

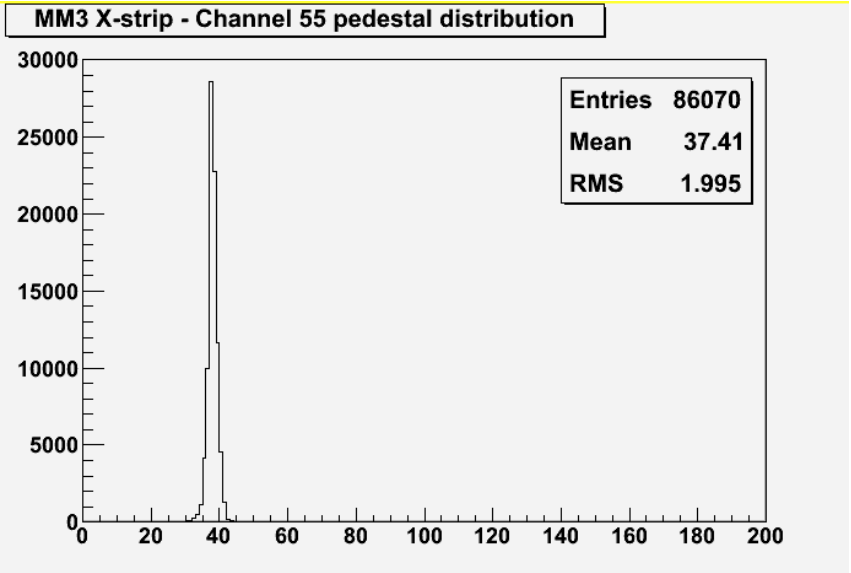


Cluster size, i.e. number of channels composing a cluster

- Cluster multiplicity, cluster size, cluster charge, cluster position are all information included in a new "reconstruction" ROOT tree, saved in a new file that is later connected to the raw data tree

Pedestal calculation

- After clustering, all channels far enough (a new free parameter to optimise!) from a cluster are added to pedestal charge histograms
  - Any improvement of clustering algorithm will automatically affect the pedestals
- Histogram are saved in the "reconstruction" file

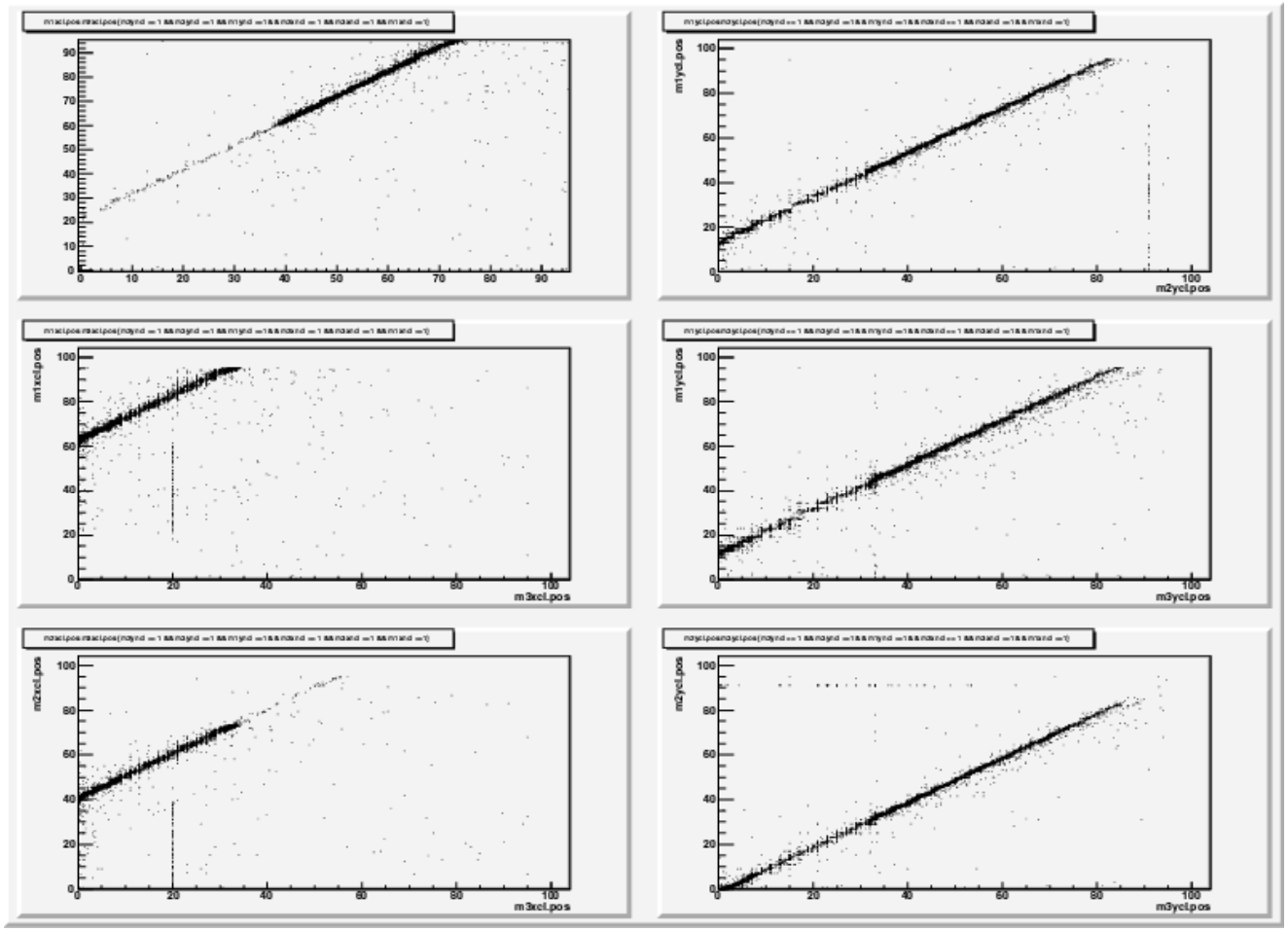


Track reconstruction

- A "Track" class has been implemented as well.
- In the following, to avoid also the strip readout combinatorial problem, tracks are reconstructed only in the events with one and only one cluster per μM station.
- Tracks are included in the "reconstruction" ROOT tree, to avoid the very long track fitting computational time (larger than 3 hours for 7000 tracks !!!)

Station misalignment problem

Cluster correlation plots between X stations



Cluster correlation plots between Y stations

- The misalignment is corrected before track reconstruction

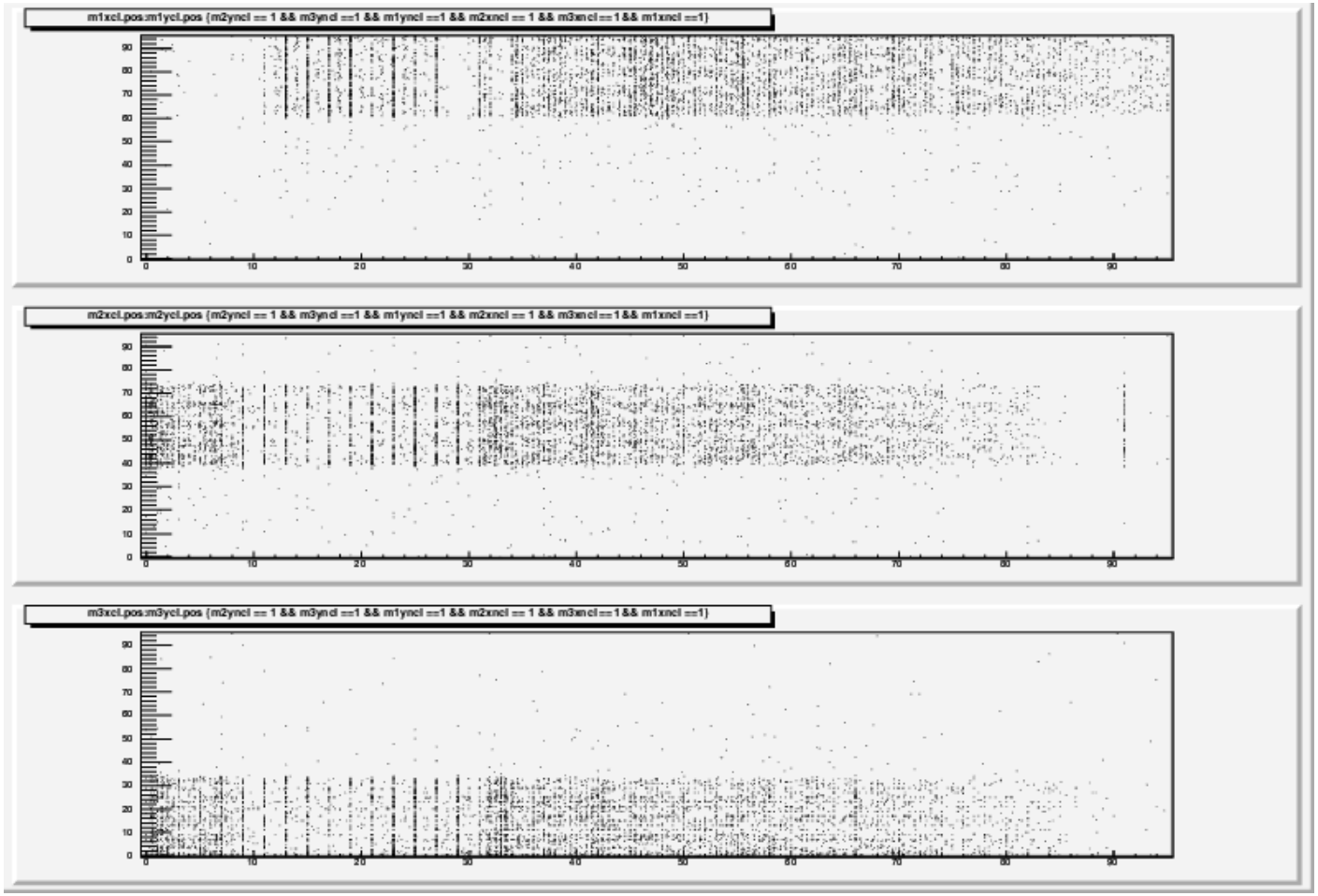
A view of the beam spot

(events with one and only one cluster per station!)

μM 1

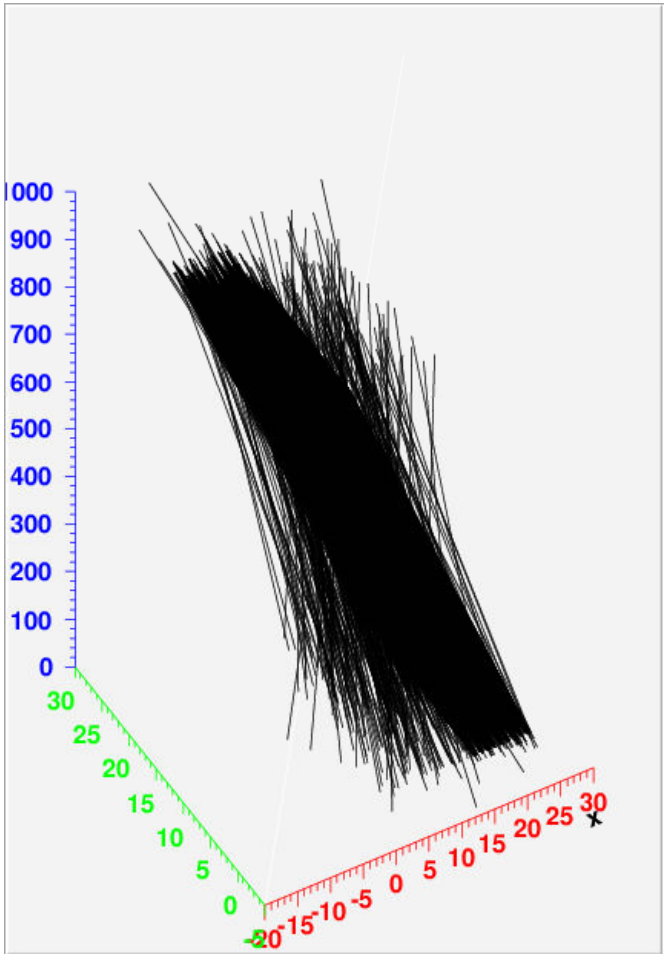
μM 2

μM 3



Reconstructed tracks

- The “reconstruction” ROOT file can be opened again for further analysis or to see the results, like with this simple “track viewer”



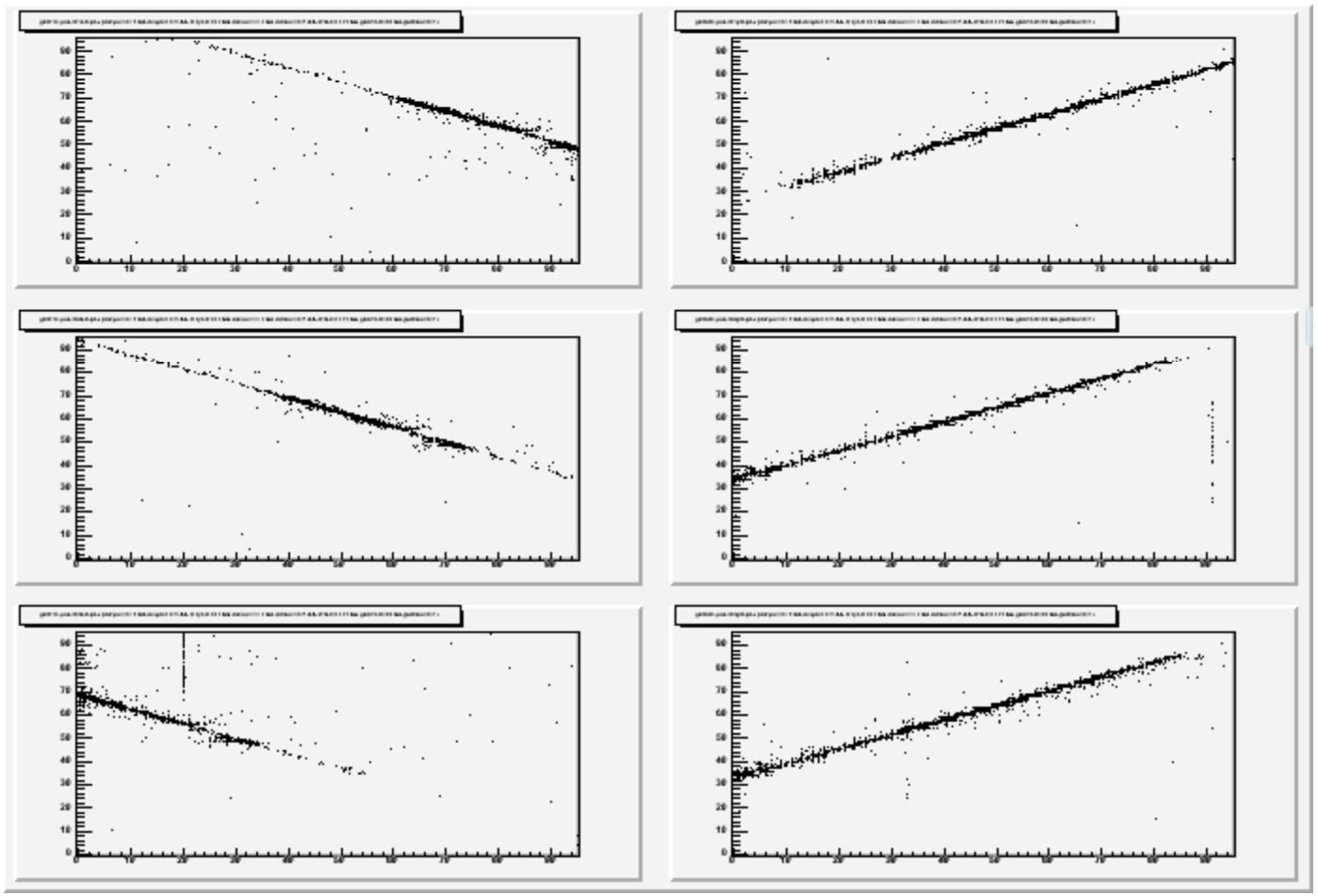
Analysis on “Device under test”

- The GEM chambers used as “Device under test” were instrumented with the same Gassiplex readout, but, due to connector mismatch, an interface board was required, leading to a mixing of channel map.
- Unfortunately we managed to resolve the problem late, so in the following slides only few correlation plots are shown.
  All the other measurements will come in the following weeks

GEM cluster vs mM stations cluster

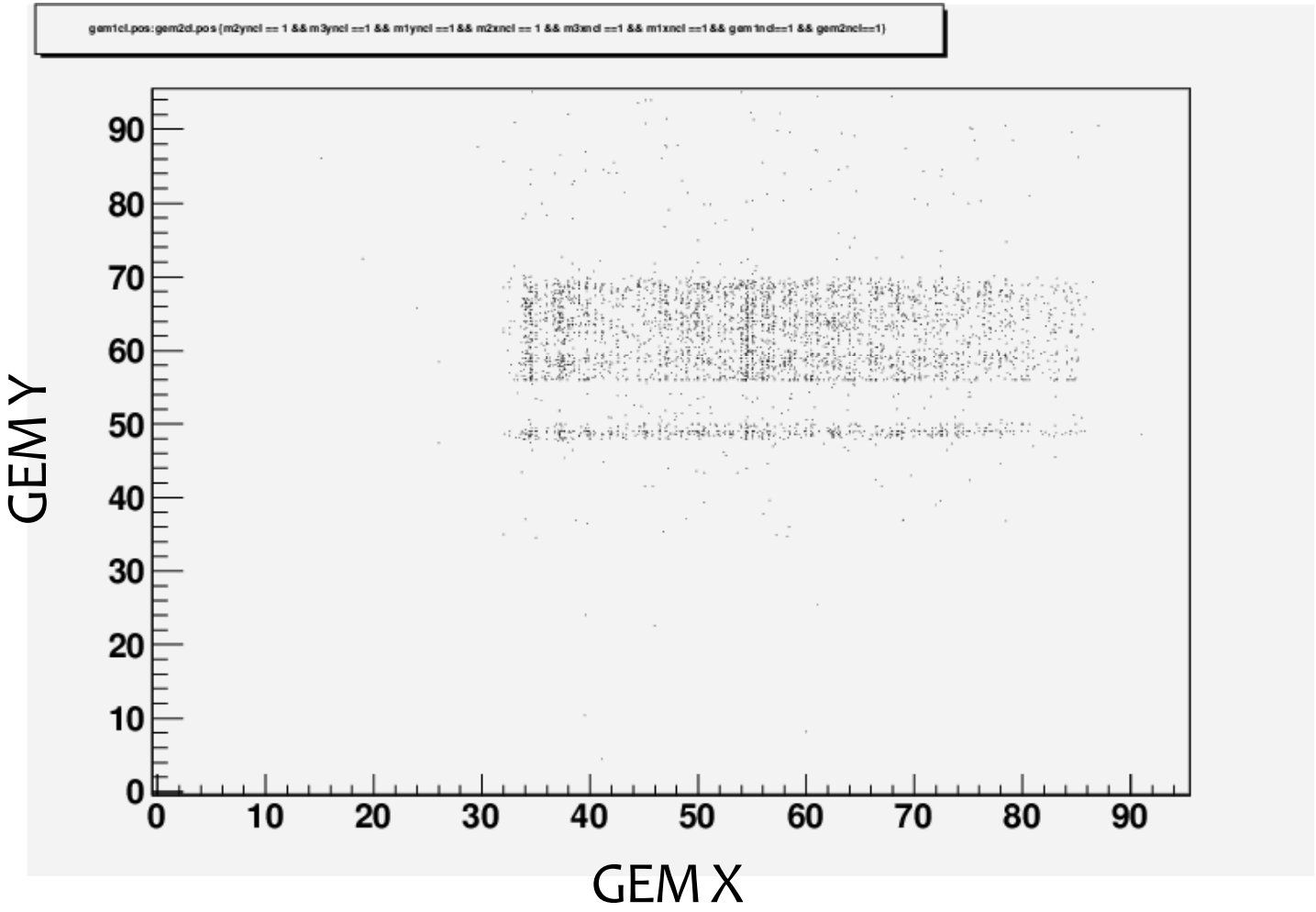
(note the different strip pitch!!!)

Cluster correlation plots between GEM X-strip and each μM X station (still numbering inversion present)



Cluster correlation plots between GEM Y-strip and each μM Y station

μM beam spot in GEM chamber



Future plans

- Complete the analysis also for the "Device under Test"
- Optimize the free parameters and improve the algorithms
- For the next October test beam, adapt the single offline analysis algorithms to online monitor algorithms