Charge signal simulation and reconstruction for 3x1x1 and 6x6x6

V. Galymov
IPNL
CERN 23.03.2017

Outline

- Raw data interface for reconstruction & analyses
- A look at muon background in 6x6x6

Library for raw data

Low-level access to raw data

Index of /WA105Soft/daq/src

Files shown: **6**

Directory revision: 366 (of 369)

Sticky Revision: [ ] Set

File ▲
Parent Directory
ChMapInterface.cc
ChannelMap.cc
EventDecoder.cc
EventEncoder.cc
HuffDataCompressor.cc
dlardaq.cc

EventDecoder.cc ← Decoder for raw event data

Interface to reading raw data (for now only for 3x1x1)

The data format produced by DAQ is described here

The low level interface to read the raw data is provided by the **EventDecoder**

Total number of channels

Number of ADC samples per ch = 10,000 for 6x6x6 and 1/6 of that for 3x1x1

Example:

```
dlardaq::EventDecoder DaqDecoder( 1280, 1667 ); // provide the size data frame
```

In addition to the event data each raw data file contains run header, run footer, and event header, which contain some useful information

dlardaq::runheader_t ← contains run number and some bytes reserved for flags
dlardaq::evheader_t ← contains event WR timestamp (+ other items)
dlardaq::footer_t ← used for internal data checks

The decompression of event data is handled automatically by EventDecoder

Headers

Run header

```
// structure to hold decoded run header
typedef struct runheader_t
{
  uint32_t run_num;
  uint8_t  run_flags;
} runheader_t;
```

Basically contains run number
+ byte reserved for flags

Event header

```
// strucutre to hold decoded event header
typedef struct evheader_t
{
  trigger_t trig_info; // trigger info
  uint8_t  dq_flag;    // data quality flag
  uint32_t ev_num;     // event number
  uint32_t ev_size;    // size of event in bytes
} evheader_t;
```

Contains trigger information
Data quality bits + compression flag
Event number
Event size (useful for compressed events)

```
typedef struct trigger_t
{
  uint8_t type;
  uint32_t num;
  struct timespec ts; //{ time_t ts.tv_sec, long tv_nsec }
} trigger_t;
```

Event timestamp

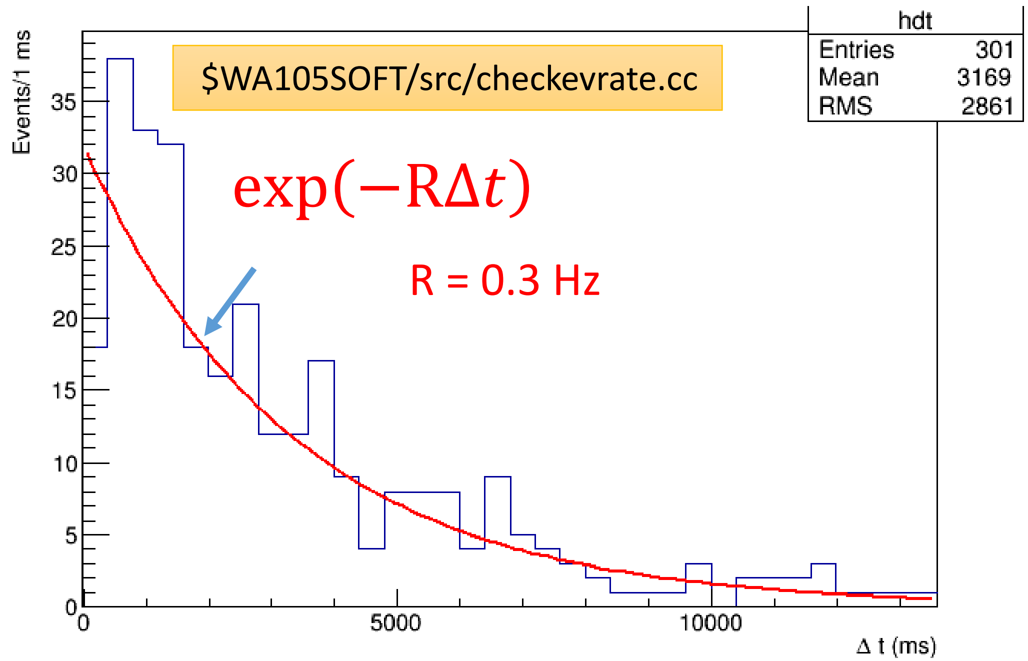
The trigger part of event header contains
Trigger type flag
Trigger number
Trigger time from WR stored in unix *timespec* structure

Example low-level event loop

```
dlardaq::EventDecoder DaqDecoder( 1280, 1667 );
// 1. Open file
DaqDecoder.Open(“blah.dat”);
// 2. Get number of events in the file
size_t nev = DaqDecoder.GetTotEvents();
// 3. Loop over all events
for(size_t i=0;i<nev;i++)
  {
   // 4. get event data
   dlardaq::evheader_t evh;
   std::vector<dlardaq::adc16_t> adc;
   DaqDecoder.GetEvent( i, evh, adc );

   // do some data manipulations …
   // example: print event header
   dlardaq::msg_info<<">>>>> Read event from file "<<i<<" / "<<nev<<endl;
   dlardaq::msg_info<<">>> Event number "<<evh.ev_num<<endl;
   dlardaq::msg_info<<">>> Event size   "<<evh.ev_size<<endl;
   dlardaq::msg_info<<">>> Trig number  "<<evh.trig_info.num<<endl;
   dlardaq::msg_info<<">>> Time stamp   "<<evh.trig_info.ts.tv_sec<<" s "<<evh.trig_info.ts.tv_nsec<<" ns"<<endl;
}
```

Δt between successive CRT trig events

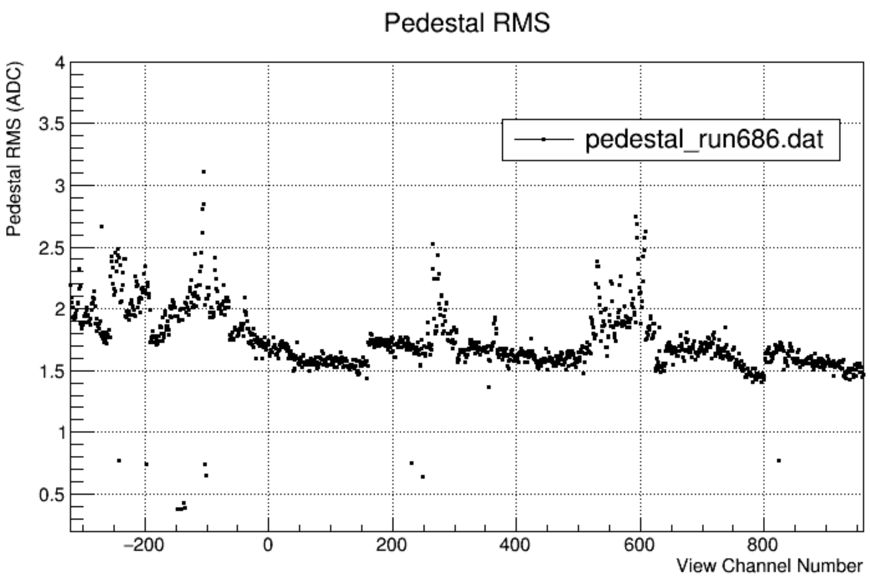


The raw data is 1D array containing ADC values ch-by-ch card-by-card
➔ Need to translate into CRP coordinates for reconstruction

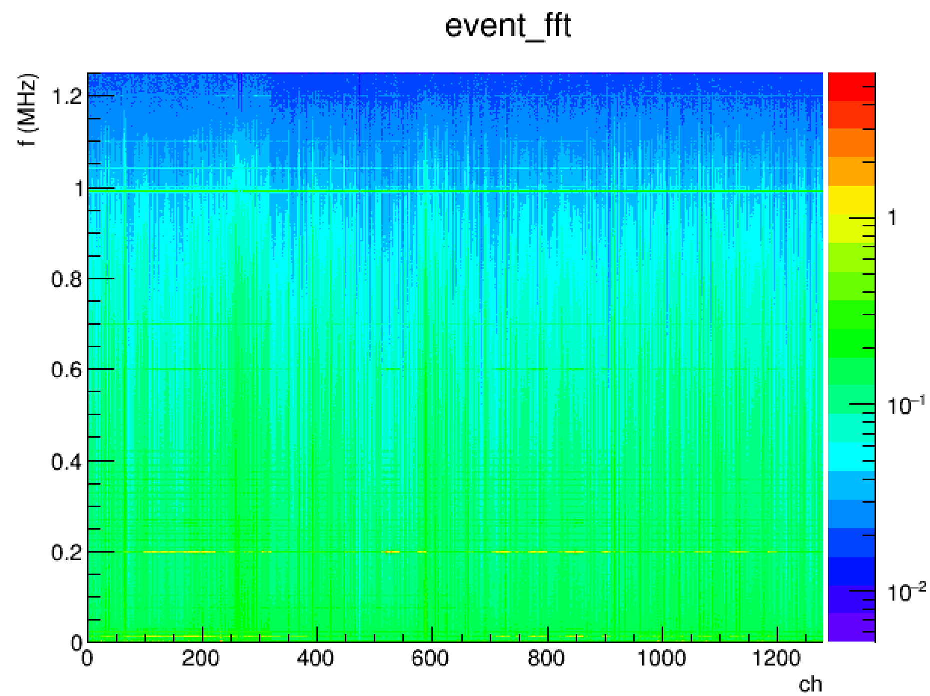
More examples of low-level event loops

Index of /WA105Soft/src

- doffft.cc
- caliana.cc



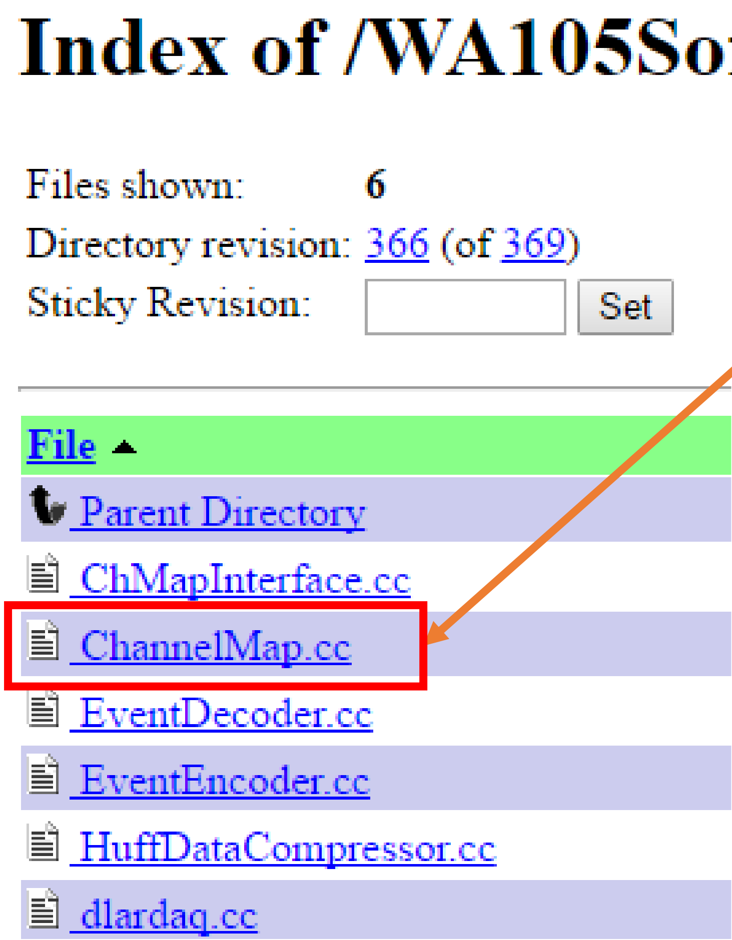
src/doffft.cc performs FFT on the raw data



src/caliana.cc (more involved) used to analyze 3x1x1 calibration data: e.g., pedestals

Library for raw data

Translating daq channel numbering into CRP coordinates



Index of /WA105Soft/daq/src

Files shown: 6
Directory revision: 366 (of 369)
Sticky Revision: [ ] Set

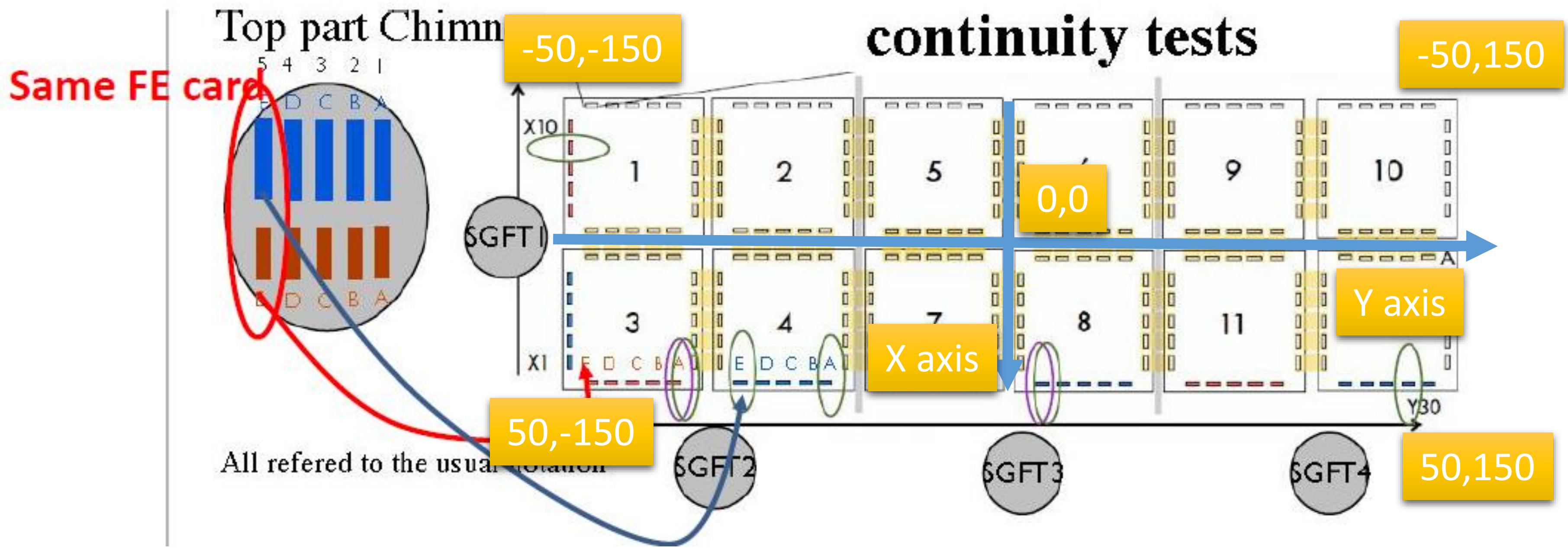
File ▲
Parent Directory
ChMapInterface.cc
ChannelMap.cc
EventDecoder.cc
EventEncoder.cc
HuffDataCompressor.cc
dlardaq.cc

DAQ channel mapping for 3x1x1 is parametrized in ChannelMap311 object

Integrate raw data with existing machinery structure in a transparent way (i.e., the software or user should not care if one runs on MC or raw data files)

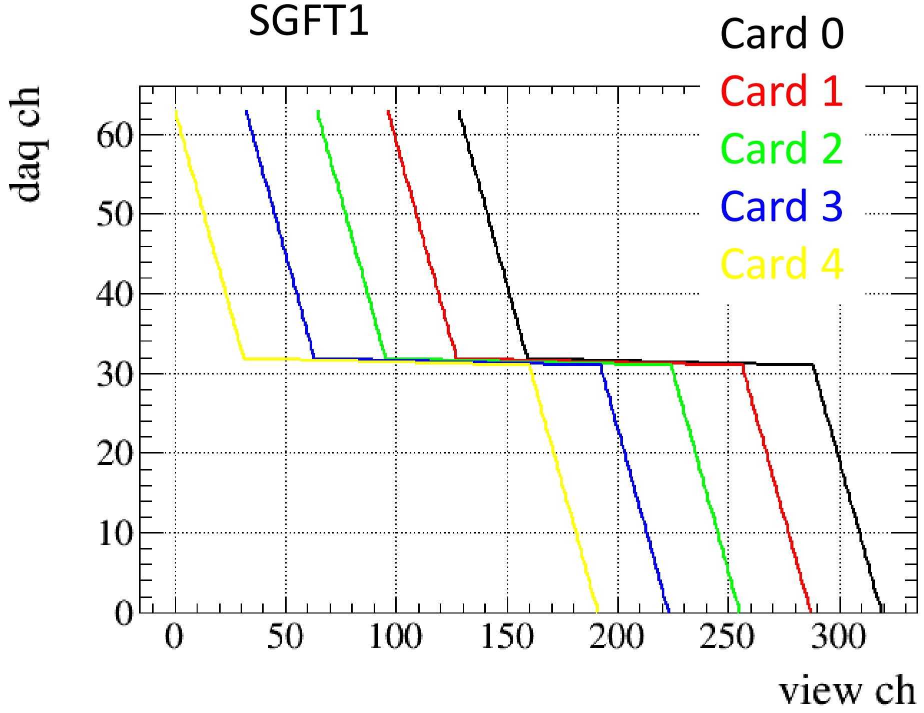
- Event display
- Reconstruction / analysis

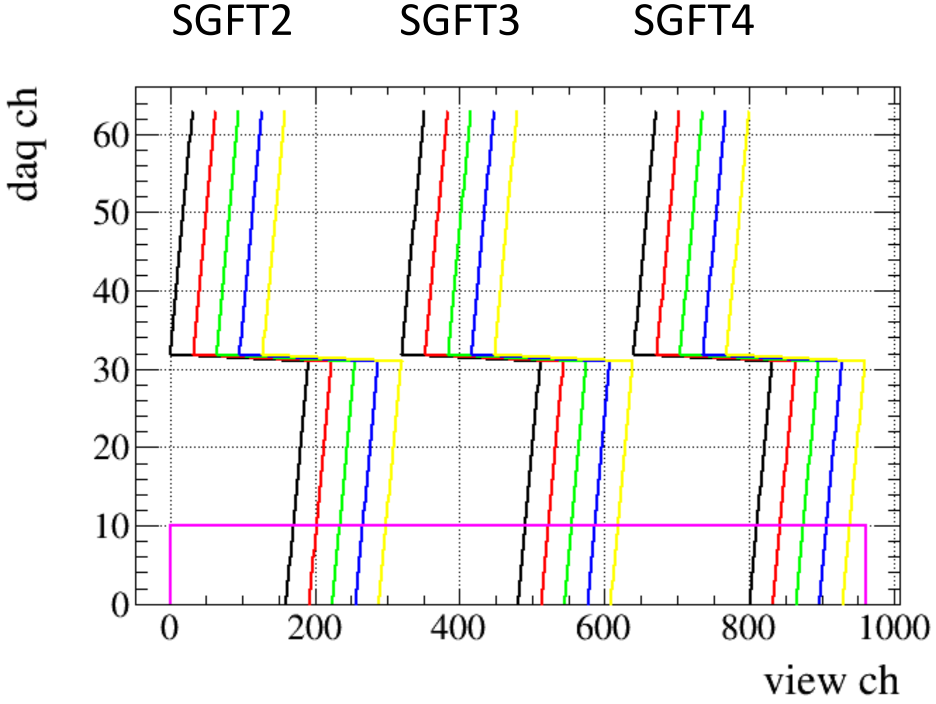
Mapping DAQ channels to “view” channels: example from 3x1x1



Coordinate system for reco/analysis needs to be **right-handed**
This defines the mapping to “view” channels:
0 – 319 for X (view 0) starting from X10 ch 32 to X1 ch 0
0 – 959 for Y (view 1) starting from Y1 ch 0 to Y30 ch 32

Graph of mapping function: 3x1x1 case





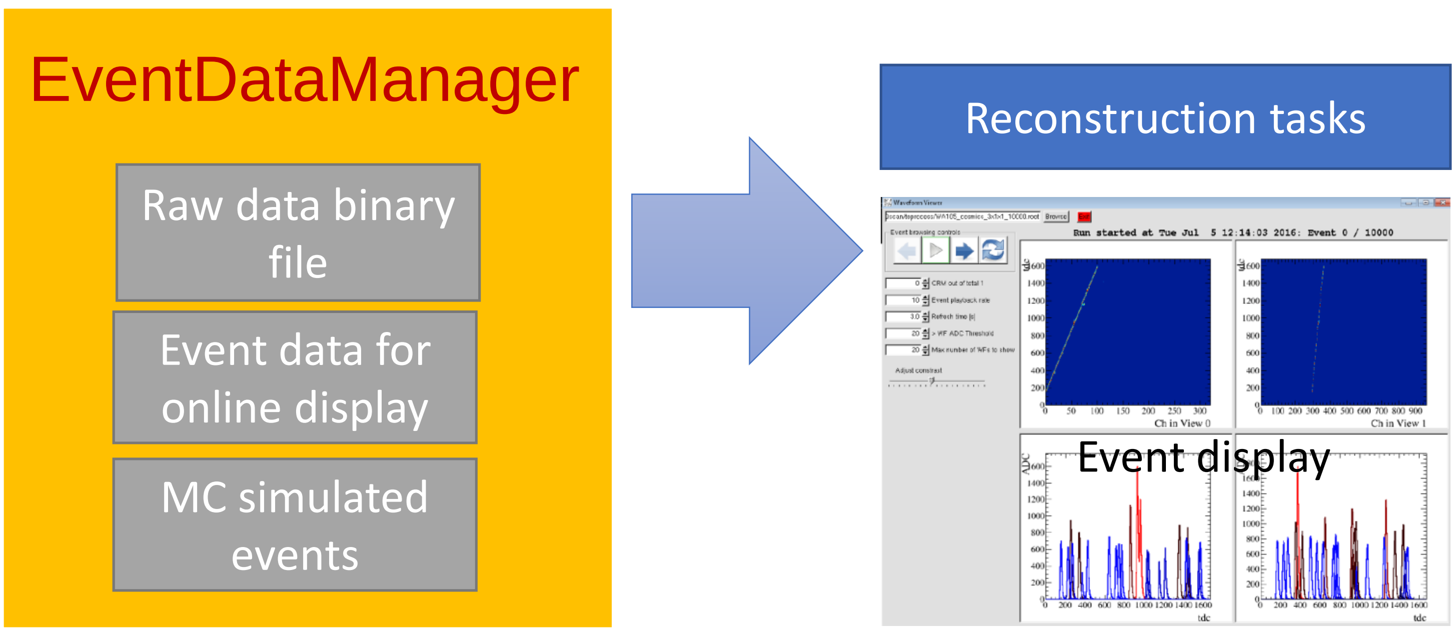
The jumps are due to 2 32-ch connectors of one card being connected on the 3x1x1 CRP 0.5 m apart (see previous page)

Unique mapping ➔ the lines should never cross!

For 6x6x6 will also need to create a channel mapping, but each cards should read 64 consecutive channels in a given view in this case

Raw data interface

- Integrate raw data with existing machinery structure in a transparent way (i.e., the software or user should not care if one runs on MC or raw data files)
  - Event display
  - Reconstruction / analysis



Propagating header information

The information contained in the headers from the DAQ raw data files is mapped to the RunHeader/EventHeader objects in WA105Soft to be available at the level of recon & analysis

e.g., EventHeader

```
Int_t     EH_EventNumber;      /* event number */
TTimeStamp EH_EventTime;       /* time stamp   */
Bool_t    EH_DQFlag;           /* event data quality flag */
```

EH_EventTime – encapsulates the WR event timestamp
EH_DQFlag – flag to signal event quality (no errors, all cards send data …)

Calibration information

WA105Soft/calidaq

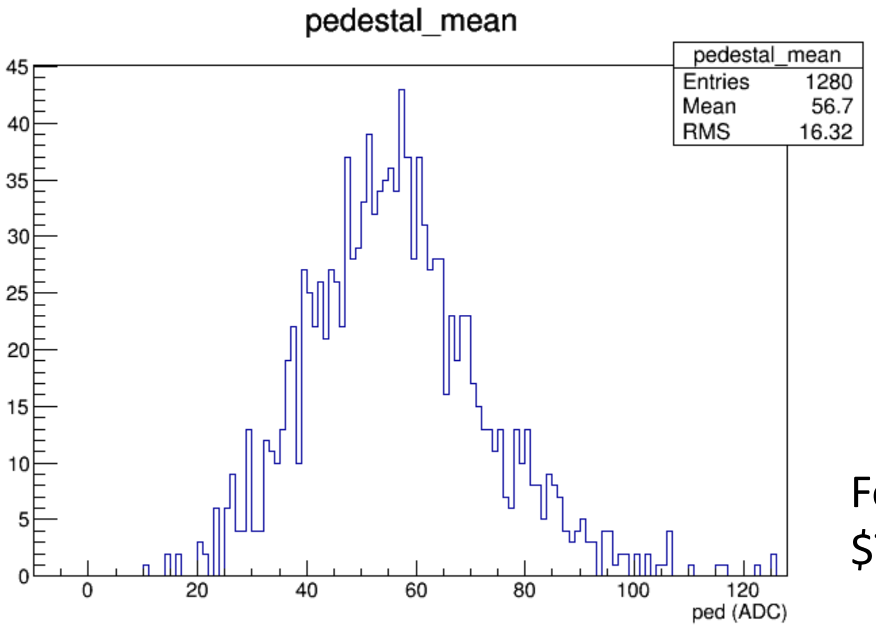
- In the case of raw data analyses:
  - ChMask ← list of non-functioning channels for reconstruction to skip
  - ChPedestal ← pedestal values and RMS measured during pedestal runs

- Calibration information is read from env path THEDATAFILES
  - E.g., ChPedestal file: $THEDATAFILES/pedestals/pedestals

To get pedestal for a given CRP (=0 for 3x1x1),
in view (0 or 1), and view channel:

```
double ped = calidaq::ChPedestal::Instance(). Pedestal( icrm, iview, ichv, pedrms )
```

Pedestal subtraction: 3x1x1 case



The mean value of the pedestal is ~56ADC with a variation of ~16ADC

- Subtract for event display to show uniform background
- For hit reconstruction to work properly also need to perform subtraction

For event display, no actions necessary provided $THEDATAFILES/pedestals/pedestals is found

In the config of recon tasks

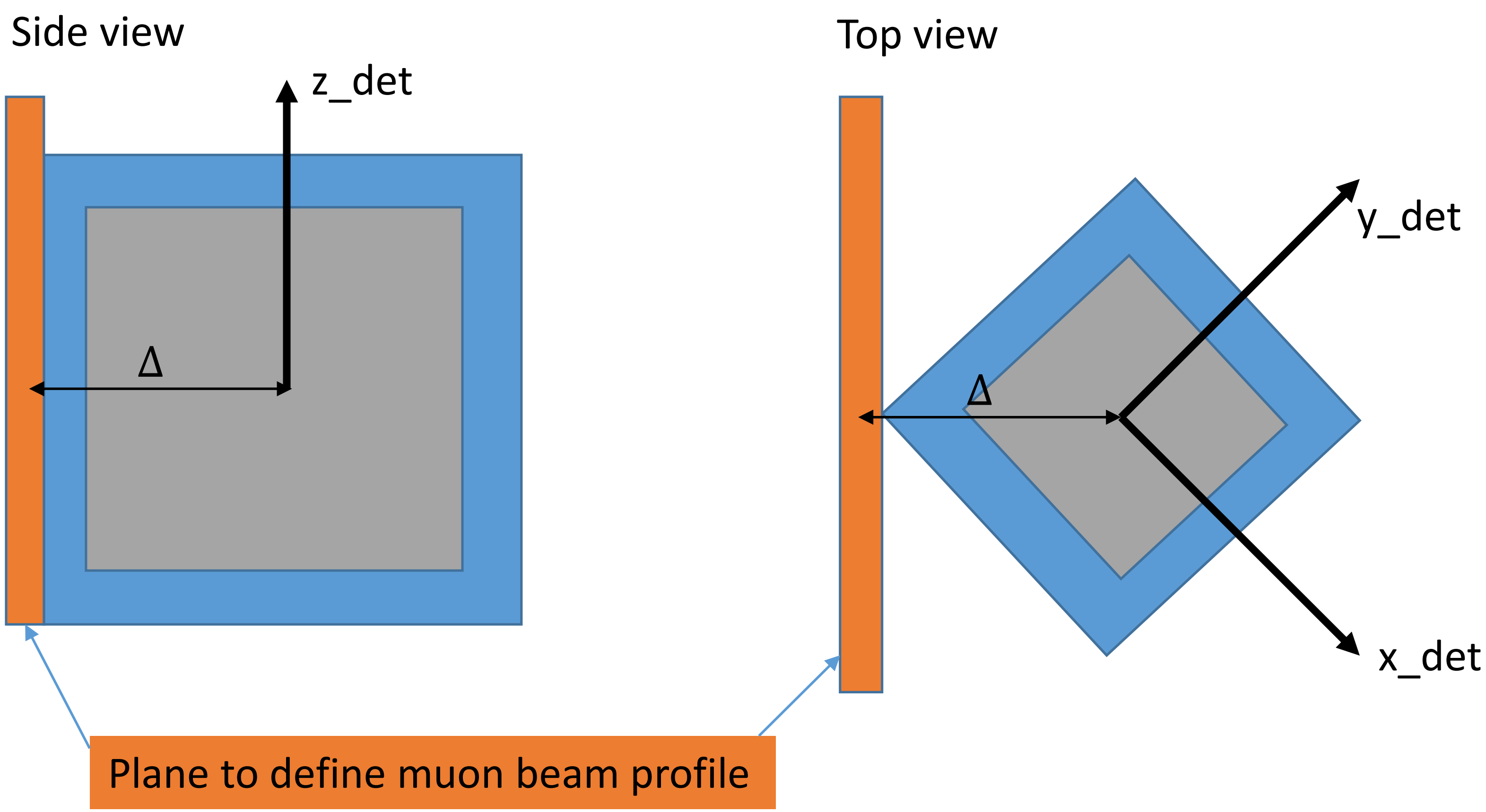
```
RECO_HIT
BEGIN_ALGORITHM 1 AlgoMultiHit
#   parameters for multihit algo
# enable using calibration info: ch masks, pedestals, calibrations
PARA_CALI [0, 1, 0]
#       relTh1 relTh2          absTh1 abs_Th2 dt(us) padLeft padRight  //
PARA_ROI [20., 4.0, 0.5, -0.1, 5, 10]        //ROI
PARA_HIT [2.5, 2.5, 9999, 9999, 3.0, 10, 10]
END_ALGORITHM
```

Enables pedestal subtraction (need $THEDATAFILES/pedestals/pedestals)

Simulation of muon backgrounds in 6x6x6

- Simulation of cosmic rays: mechanism to simulate has been implemented for several years
- Simulation of muon halo: recently added

Simulation of muon halo: geometry



Profile definition in plane

y, y' -- position and angle along a given coordinate in the reference given plane
ϵ – emittance in units of [cm mrad]
α – Twiss parameter (dimensionless) if 0 no correlation between angle and position
$\bar{y}, \bar{y}'$ -- beam position and angle in the reference plane

$$B(y, y'; \bar{y}, \bar{y}', \sigma_y, \epsilon_y, \alpha_y) = \frac{1}{2\pi\sigma_y\sigma_{y'}\sqrt{1-\rho_{yy'}^2}} \exp\left(-\frac{1}{2(1-\rho_{yy'}^2)}\mathcal{F}\right)$$

with

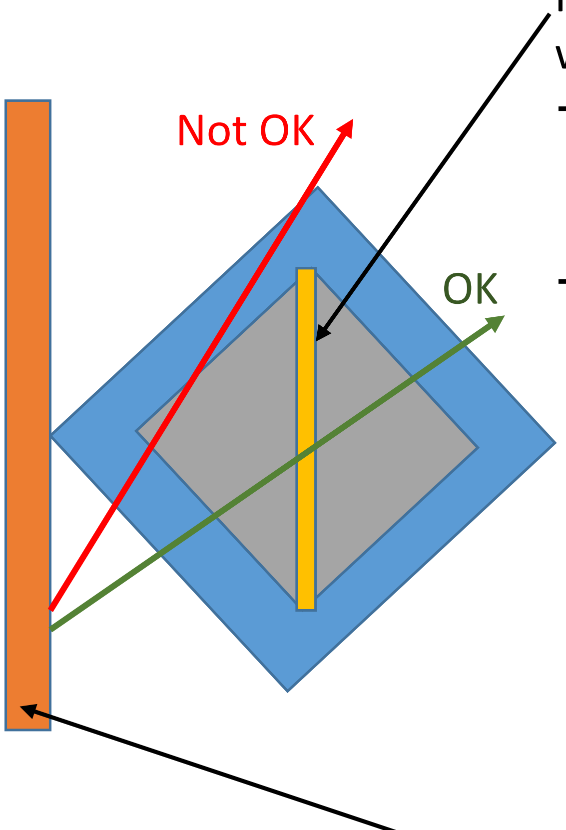
$$\mathcal{F} = \frac{(y-\bar{y})^2}{\sigma_y^2} + \frac{(y'-\bar{y}')^2}{\sigma_{y'}^2} - \frac{2\rho_{yy'}(y-\bar{y})(y'-\bar{y}')}{\sigma_y\sigma_{y'}},$$

where

- $\sigma_{y'} = \epsilon_y\sqrt{1+\alpha_y^2}/4\sigma_y$
- $\rho_{yy'} = -\alpha_y/\sqrt{1+\alpha_y^2}.$

The area of the beam in y-y' plane the area of an ellipse given by $A = \epsilon\pi$

MC procedure



Intersection with this plane $600\sqrt{2} \times 600$ cm2 defines whether to accept or reject the randomly drawn vector

→ not general enough for all direction, but steep angles should not be realistic I would think (more generally could check for the intersection with the two front faces)

→ The plane coordinates wrt BHP is (800 cm, 0, 0)

If the direction is accepted, the coordinates are transformed into detector coordinate system and the muon is added to the MC transport stack

Currently no momentum distribution is assumed (easy to include though) and starting momentum is set to 10GeV

The beam profile is defined in this plane (Beam Halo Plane or BHP) and random vector are drawn according to the specified multi-Gaussian distribution

Enabling of simulation of muon halo

Similar to the CR background simulation need to set in the input card
IFILE WA105_MCEVENT

Then specify the parameters for muon beam halo (NB: this just an example, proper parameter values should be obtained from the beamline simulation)

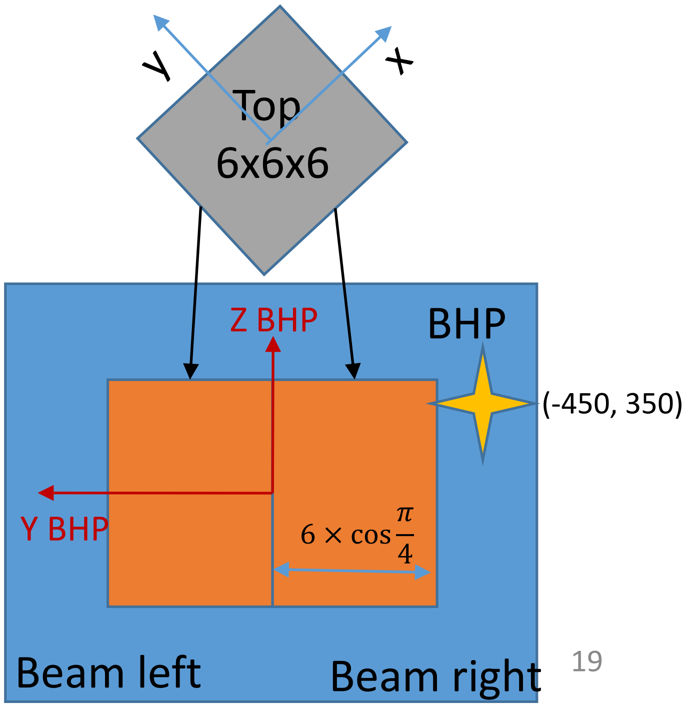
```
# beam halo properties
WA105MC_BH 5.0E+4 -450.0 0.0 30.0 40.0 0.0 350.0 0.0 30.0 40.0 0.0
# time window
WA105MC_BHWIN -4.0 8.0
```

← Start time and duration in ms

For this set of parameters particles in the active volume is 8kHz

The card WA105MC_BH parameters are:

1. Rate in Hz
2. Y0 in Beam Halo Plane (BHP) in cm
3. ThetaY0 in (BHP) in mrad
4. Sigma in Y in cm
5. Emittance in Y in cm x mrad
6. Twiss alpha in Y
7. Z0 in cm
8. ThetaZ0 in mrad
9. Sigma in Z in cm
10. Emittance in Z in cm x mrad
11. Twiss alpha in Z



The coordinate system is a bit odd, but this is to simplify transformations to the coordinate system of detector geometry

Enabling of simulation of muon halo

Similar to the CR background simulation need to set in the input card
IFILE WA105_MCEVENT

Then specify the parameters for muon beam halo (NB: this just an example, proper parameter values should be obtained from the beamline simulation)

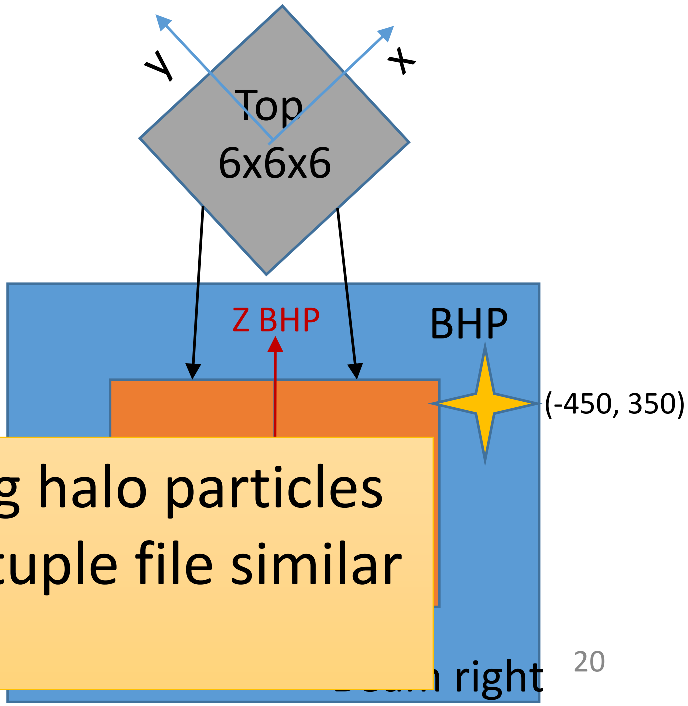
```
# beam halo properties
WA105MC_BH 5.0E+4 -450.0 0.0 30.0 40.0 0.0 350.0 0.0 30.0 40.0 0.0
# time window
WA105MC_BHWIN -4.0 8.0
```

← Start time and duration in ms

For this set of parameters particles in the active volume is 8kHz

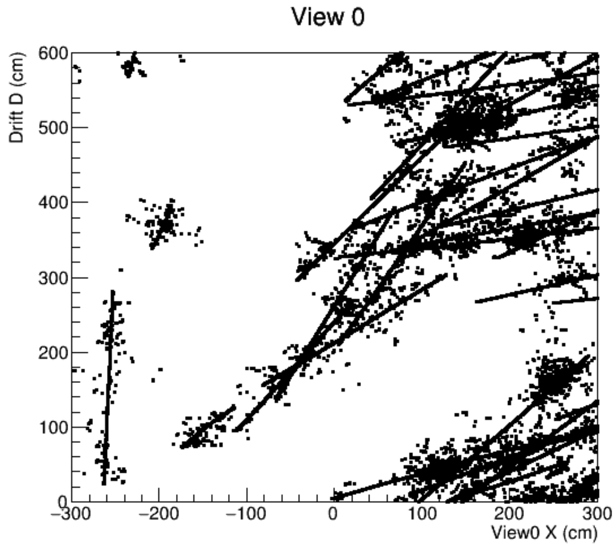
The card WA105MC_BH parameters are:

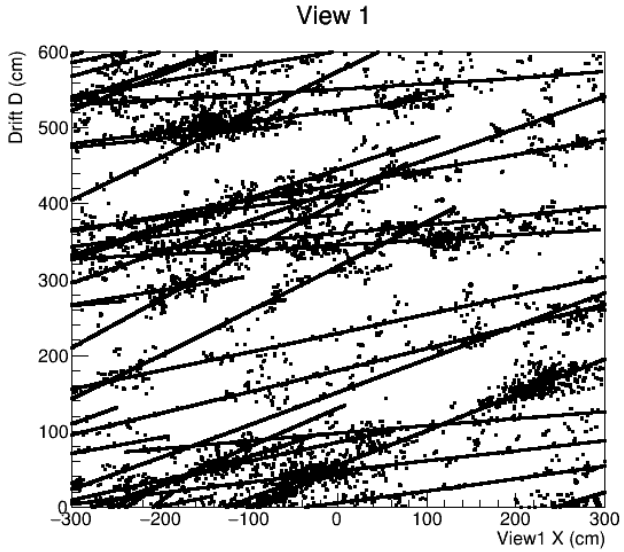
1. Rate in Hz
2. Y0 in Beam Halo Plane (BHP) in cm
3. ThetaY0 in (BHP) in mrad
4. Sigma in Y in cm
5. Emittance in Y in cm x mrad
6. Twiss alpha in Y
7. Z0 in cm
8. ThetaZ0 in mrad

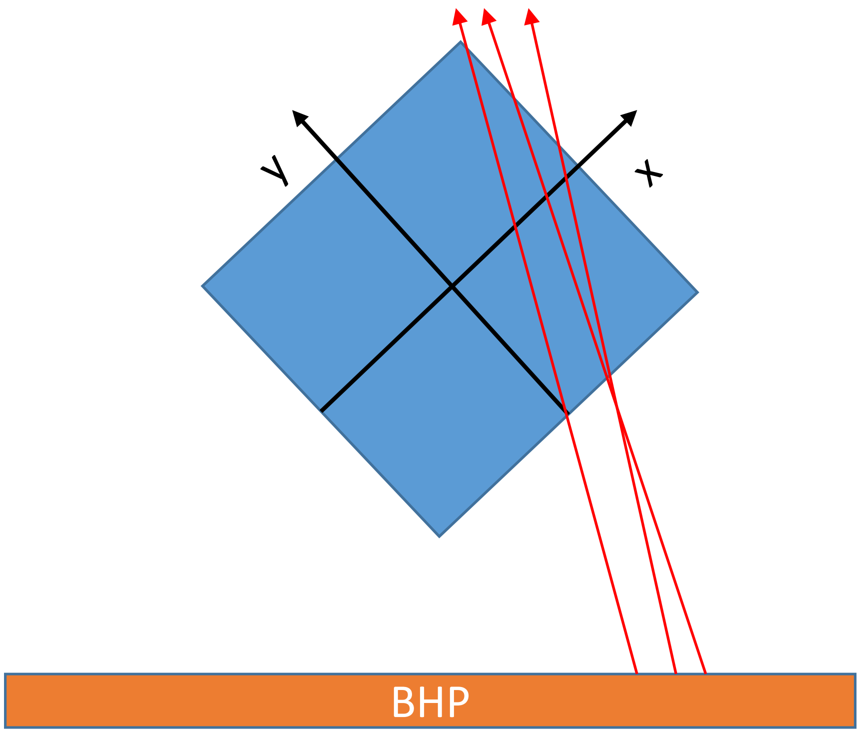


More generally, can also implement reading halo particles (ideally defined in BHP) from an external ntuple file similar to the mechanism used for cosmics

View from the CRP





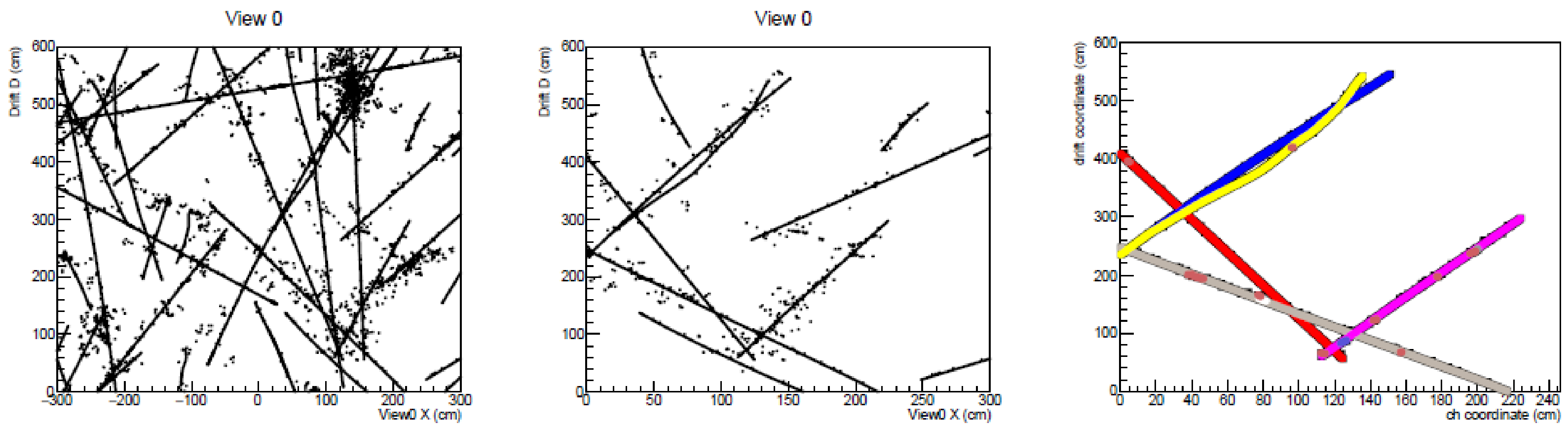


With mean beam position set on the beam right most of the particle trajectories will cross on the positive X side of the CRP

CR tracks for online analyses

- Quick hit/track reconstruction is required for online analyses on subset of data to monitor detector performance
- Purity:
  - Sufficient to have 2D tracks segments (no need to merge into 3D) ➔ could get O(10) reasonably long tracks per event
  - Measurement of relative attenuation (presentation by Elisabetta at SB), but would need T_0 (absolute muon arrival time wrt to beam trigger) if test stratification of impurities is desired
- LEM gain monitoring:
  - Need to reconstruct 3D path to get dE/dx → merging of two views is necessary
  - Need to know T_0 to apply charge attenuation correction as a function of the true drift distance
  - Match T_0 from LRO with T_0 determined from certain track topologies

Muon track reconstruction



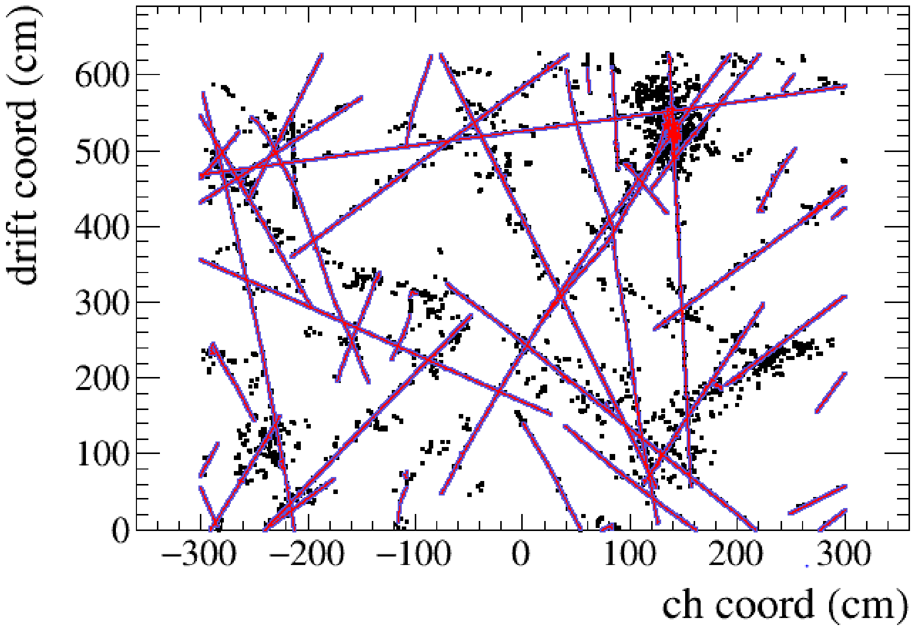
- Track segments are reconstructed in each view and in each CRP sub-module in a given disconnected sub-cluster
- A 2D Kalman Filter is used to build track trajectories
  - Includes MS effects in order to follow non-straight trajectories
- Tracks segments are then combined between all CRP units to form complete 2D track in a given view
- Tracks are combined between views based on the best match of endpoints in time and the total charge observed in each view

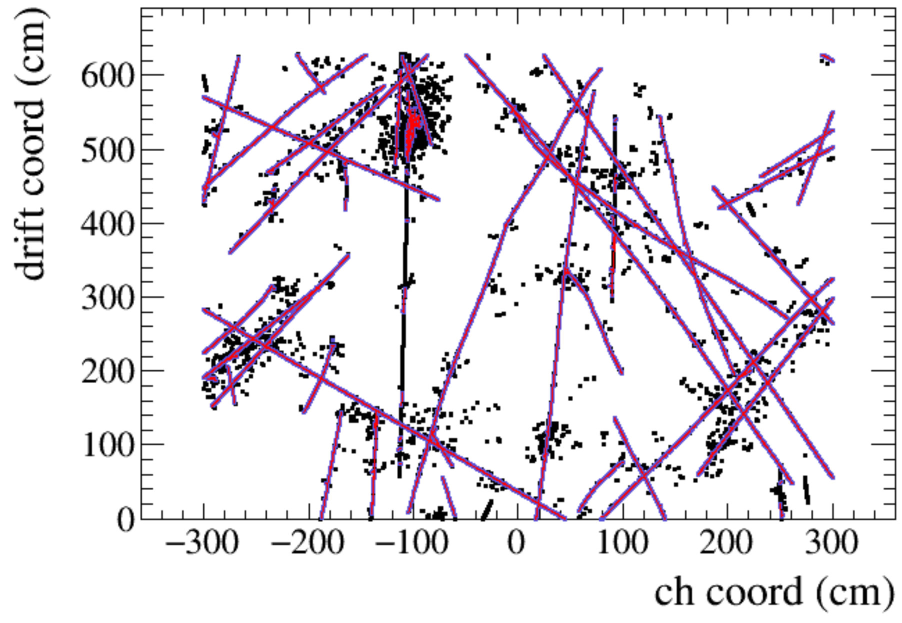
Example event (CR only)

Black points show the reconstructed hits

Magenta points show hits associated to some track

Red lines indicate track paths





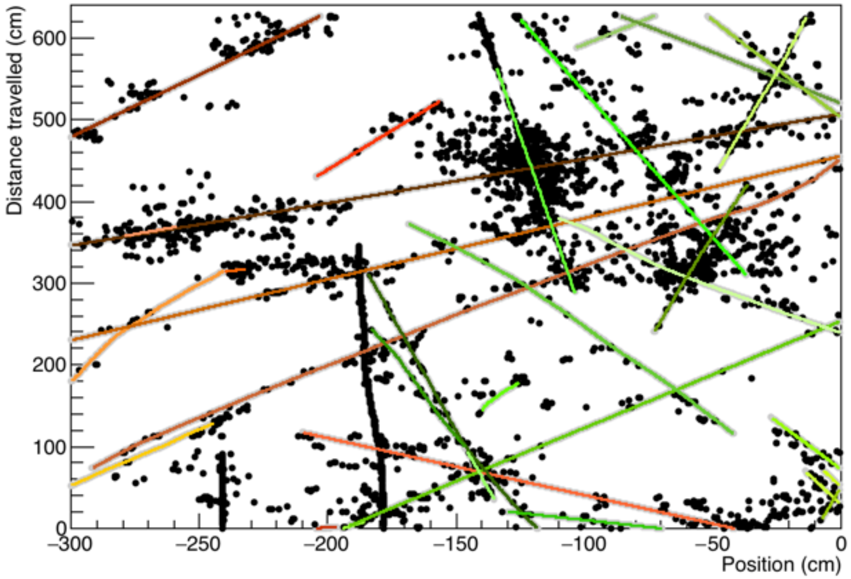
Track finder : ClusFilter

L. Zambelli

Remove clustering step to save time → ClusFilter algorithm idea is to find tracks directly at the clustering level:

1. Seed a track with 3 neighbouring hits
2. Initiate a kalman-like filter if the seeding hits are fitted with a line with good correlation coefficient
3. Neighbouring hits are added to the track if they fulfill chi2 requirement, and the filter is updated
4. The filter is in 2D, the track momentum is not fitted (would need 3D tracking)
5. Code allows to skip a few channels

Example : CR event, 1 CRM



- Hits associated to a track
- Unassociated hits

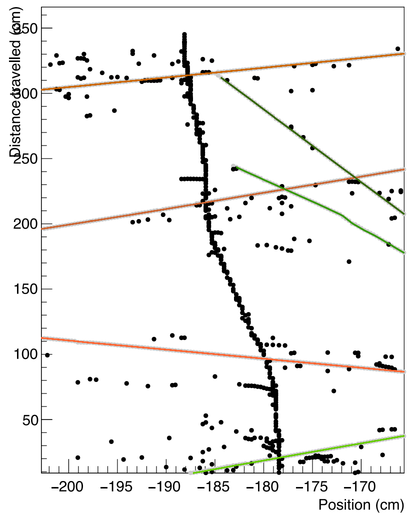
Not so many delta rays found as the minimum number of hits to make a track is set to 20 [input option]

ClusFilter

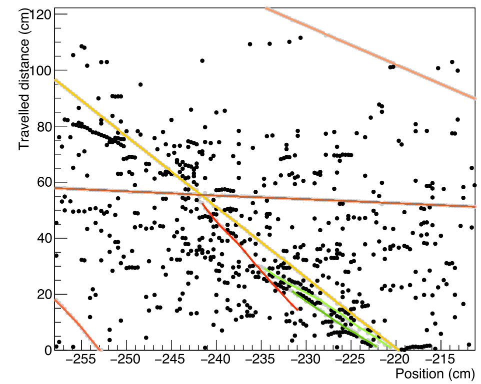
L. Zambelli

Few issues remaining

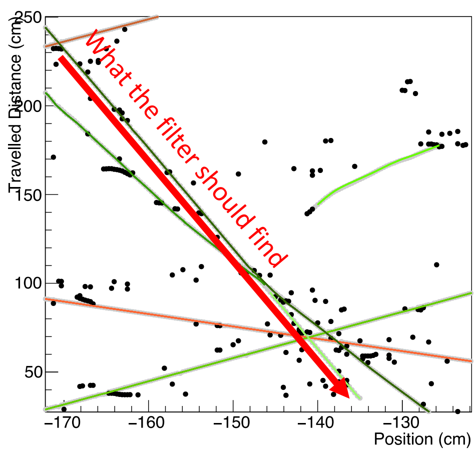
Vertical tracks are not found :



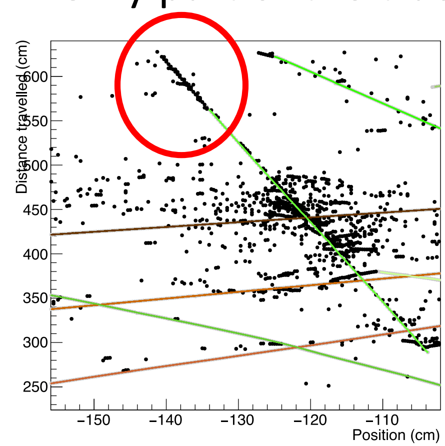
In crowded areas, random tracks can be found:



Filter gets confused when two tracks with similar slopes crosses :



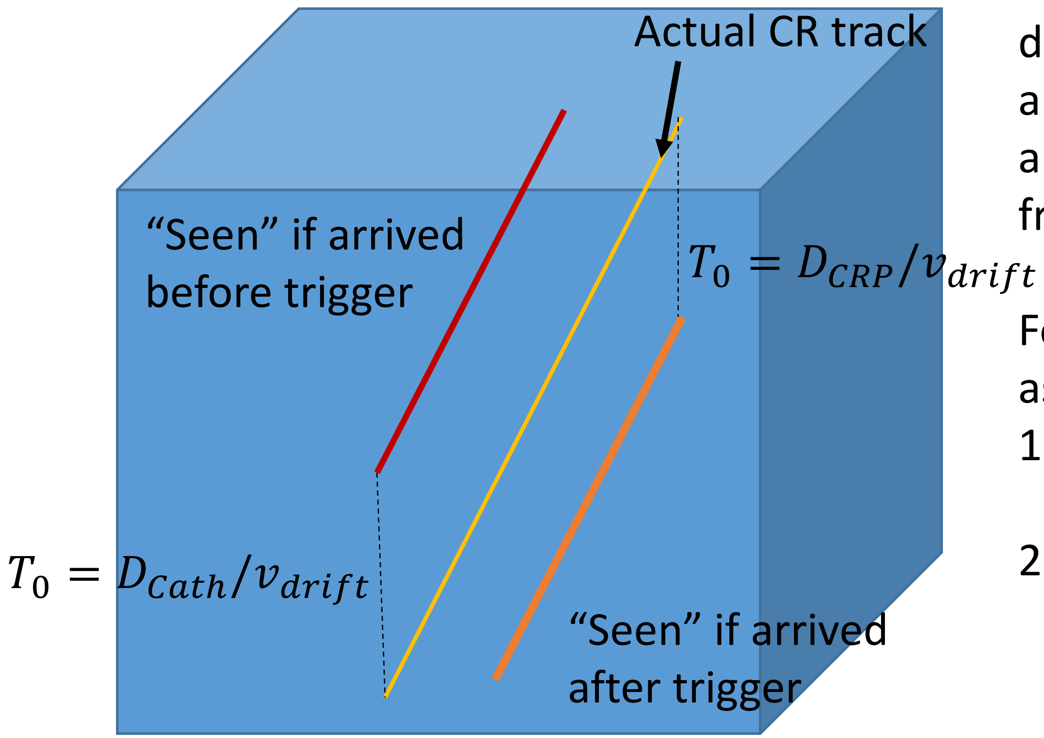
Only part of the track is found :



CR absolute arrival time from tracks

It is possible to calculate T_0:

- For CR arriving before trigger that exit on the cathode side
- For CR arriving after trigger that enter on the CRP side

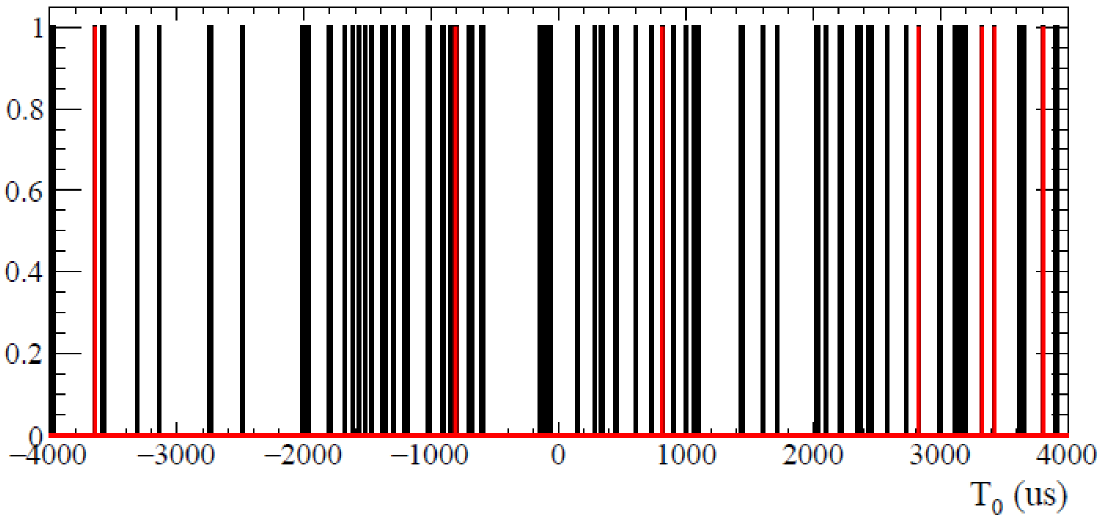


So from endpoints that appear to hang disconnected from either cathode or anode in 3D, we should be able to get T_0 and cross check it against T_0 reconstructed from PMTs

For other situations will not be possible to assign T_0

1. Will not use for gain monitoring (will not know lifetime correction)
2. Can use for purity analysis
   - As was shown, the analysis can be done for relative Q attenuation

Selecting tracks for online analysis

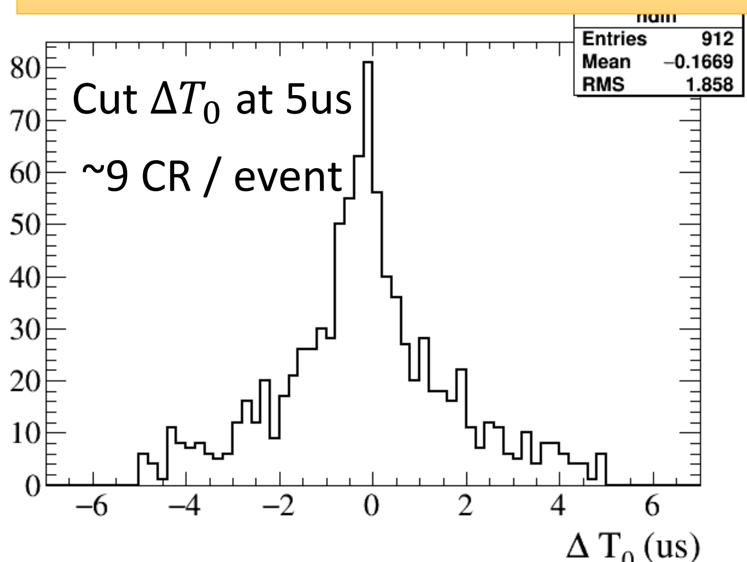


True CR arrival time
(to be updated with the measured T_0 from LRO)

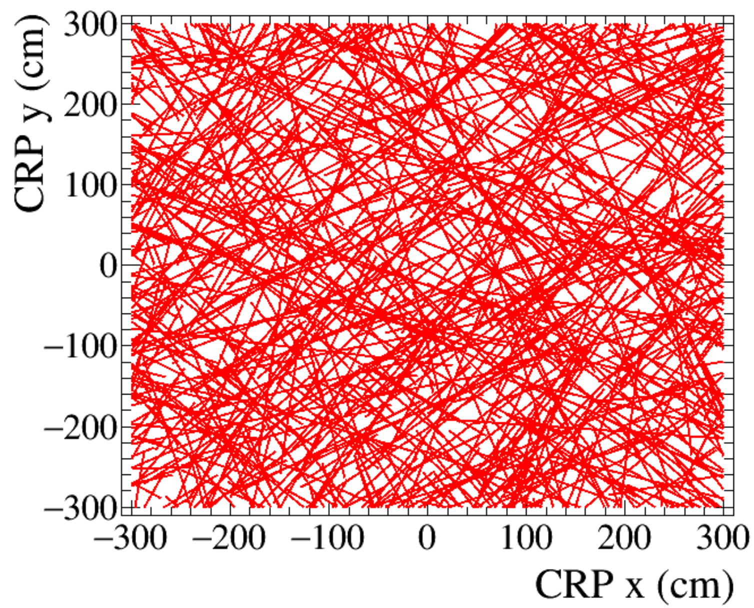
T_0 reconstructed from track end-points

Put a time cut on maximal ΔT and also a veto window around T_0 = 0

100 simulated CR background events
~1s of data taking at 100Hz trig rate



CRP Area coverage



Muon background removal

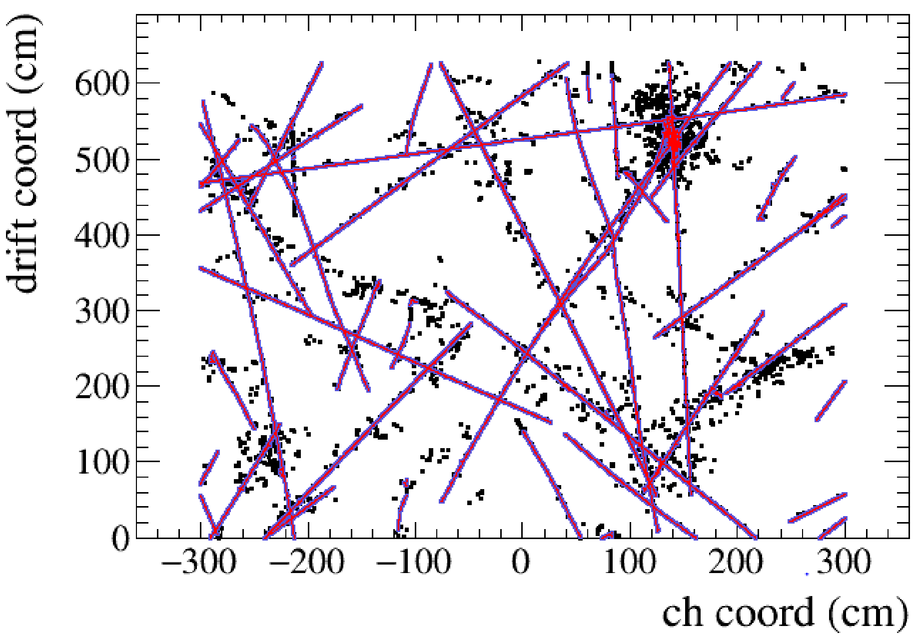
Removal is a more challenging topic

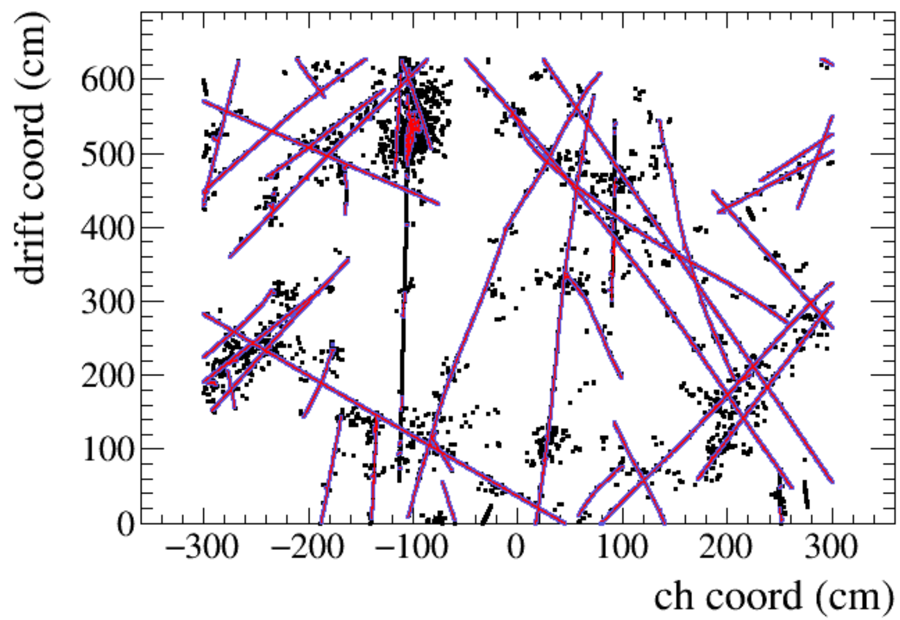
- Need to handle CRs in the presence of the beam event and how to deal with that

- Study of the correlation with reconstructed light information
  - Can get T_0 from track topology and relate that to T_0 from light signals for some cases
  - How well do we understand the number of expected background tracks in the detector given the information from the light data?

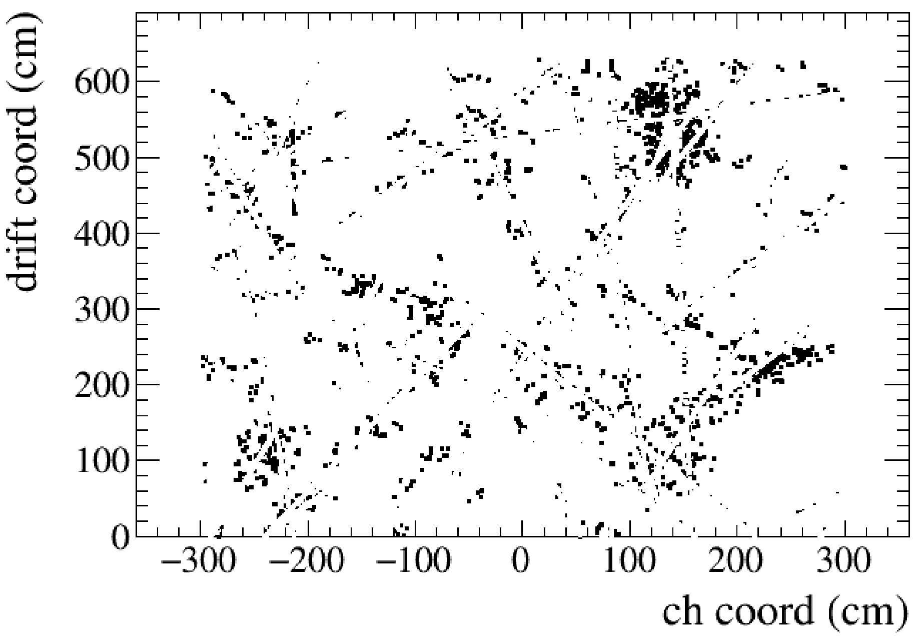
Example event (CR only)

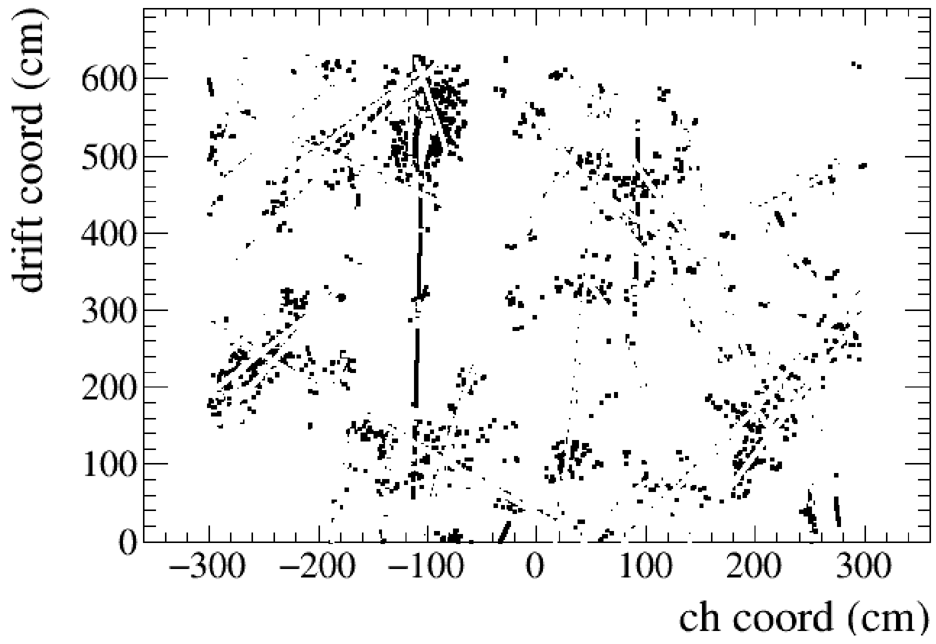
Black points show the reconstructed hits
Blue points show hits associated to some track
Red lines indicate track paths





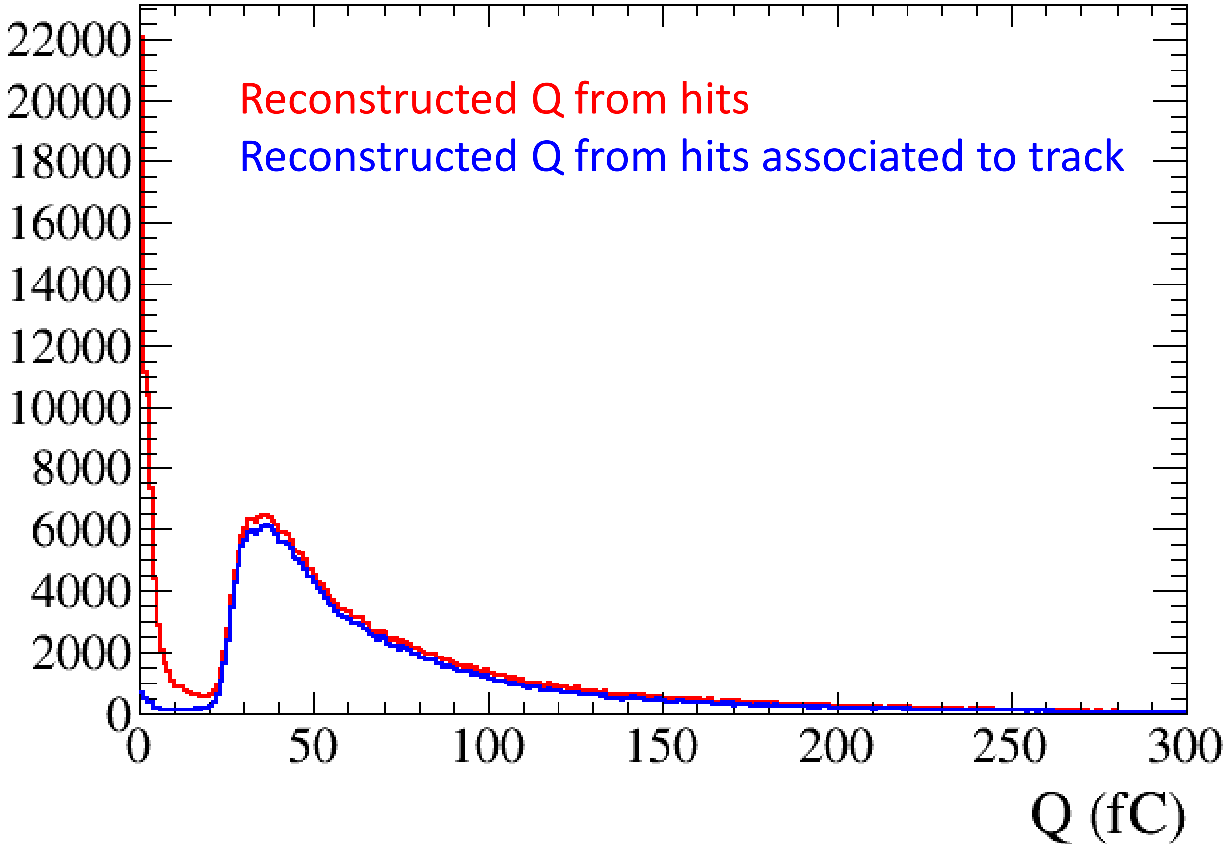
Example event – tracks removed





Event with CR only
Vertical tracks are not handled as well in reconstruction (to improve)
Showers are not treated so heavy activity near the tracks remain

Leftover charge



Most of the missing charge comes from in isolated low-q hits (brem photons)
Assuming all the tracks are identified as cosmic rays perfectly, this is still an important item for precision calorimetric measurements

In conclusion

- Framework to interface to the raw data from 3x1x1 is in place
- Work on preparing software for the 6x6x6 operation is on-going
  - Fast event reconstruction for online monitoring
  - Selection of CR tracks for online analyses to assess the detector performance

- Offline data analyses is a broader topic …
  - E.g., muon background removal is not a simple task in view of calorimetric calibration measurements