CLAS12 Tracking Overview and Progress

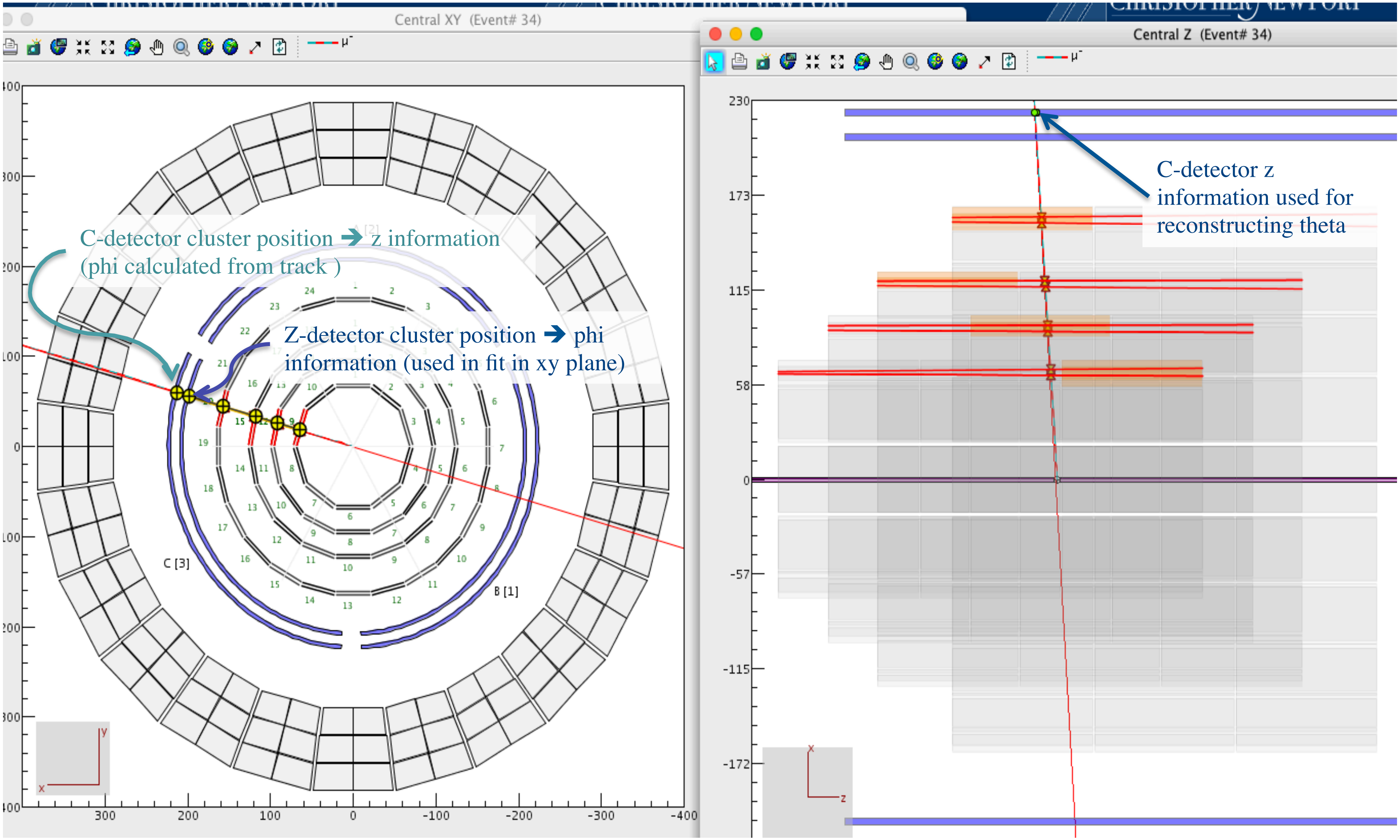
Veronique Ziegler

First Experiment Workshop

Central Vertex Tracker Reconstruction

- New package clasrec-CVT contains algorithms to reconstruct events using BMT and SVT
  - SVT can be run stand alone
  - Raw data translation for both systems
    - tested on raw data with BMT + SVT hits
  - new algorithms to use BMT cluster positions
    - improved BMT (also FMT) clustering algorithm under development (Saclay- Maxime Defurne)
  - Validation flags can be turned on → layer efficiencies
- Alignment code development using Millepede (J. Gilfoyle)
- Geometry implementation in common tools package (P. Davies)
- Ongoing validation (next slides)

BMT hit info used in pattern recognition



New Banks

BMT::dgtz | BMT::true | *BMTRec::Clusters*
BMTRec::Crosses | *BMTRec::Hits* | *BST::dgtz*
BST::true | *BSTRec::Clusters*
BSTRec::Crosses | *BSTRec::Hits*
BSTRec::Trajectory | CND::dgtz
CNDRec::hits | CTOF::dgtz
CTOFRec::ctofhits | *CVTRec::Cosmics*
CVTRec::Trajectory | DC::dgtz

3115	0	CVTRec::Cosmics	ALSOBANK	280
3115	1	CVTRec::Cosmics.ID	INT32	4
3115	2	CVTRec::Cosmics.trkline_yx_slope	DOUBLE64	8
3115	3	CVTRec::Cosmics.trkline_yx_interc	DOUBLE64	8
3115	4	CVTRec::Cosmics.trkline_yz_slope	DOUBLE64	8
3115	5	CVTRec::Cosmics.trkline_yz_interc	DOUBLE64	8
3115	6	CVTRec::Cosmics.theta	DOUBLE64	8
3115	7	CVTRec::Cosmics.phi	DOUBLE64	8
3115	8	CVTRec::Cosmics.chi2	DOUBLE64	8
3115	9	CVTRec::Cosmics.ndf	INT32	4
3115	25	CVTRec::Cosmics.Cross1_ID	INT32	4
3115	26	CVTRec::Cosmics.Cross2_ID	INT32	4
3115	27	CVTRec::Cosmics.Cross3_ID	INT32	4
3115	28	CVTRec::Cosmics.Cross4_ID	INT32	4
3115	29	CVTRec::Cosmics.Cross5_ID	INT32	4
3115	30	CVTRec::Cosmics.Cross6_ID	INT32	4
3115	31	CVTRec::Cosmics.Cross7_ID	INT32	4
3115	32	CVTRec::Cosmics.Cross8_ID	INT32	4
3115	33	CVTRec::Cosmics.Cross9_ID	INT32	4
3115	34	CVTRec::Cosmics.Cross10_ID	INT32	4
3115	35	CVTRec::Cosmics.Cross11_ID	INT32	4
3115	36	CVTRec::Cosmics.Cross12_ID	INT32	4
3116	0	CVTRec::Trajectory	ALSOBANK	1440
3116	1	CVTRec::Trajectory.ID	INT32	80
3116	2	CVTRec::Trajectory.LayerTrackIntersPlane	INT32	80
3116	3	CVTRec::Trajectory.SectorTrackIntersPlane	INT32	80
3116	5	CVTRec::Trajectory.XtrackIntersPlane	DOUBLE64	160
3116	6	CVTRec::Trajectory.YtrackIntersPlane	DOUBLE64	160
3116	7	CVTRec::Trajectory.ZtrackIntersPlane	DOUBLE64	160
3116	8	CVTRec::Trajectory.PhiTrackIntersPlane	DOUBLE64	160
3116	9	CVTRec::Trajectory.ThetaTrackIntersPlane	DOUBLE64	160
3116	10	CVTRec::Trajectory.trkToMPlnAngl	DOUBLE64	160
3116	11	CVTRec::Trajectory.CalcCentroidStrip	DOUBLE64	160

CVTRec → reconstruction banks using both BMT + SVT

CVT Offline Monitoring and Validation

Y. Gotra

CVT Validation suite

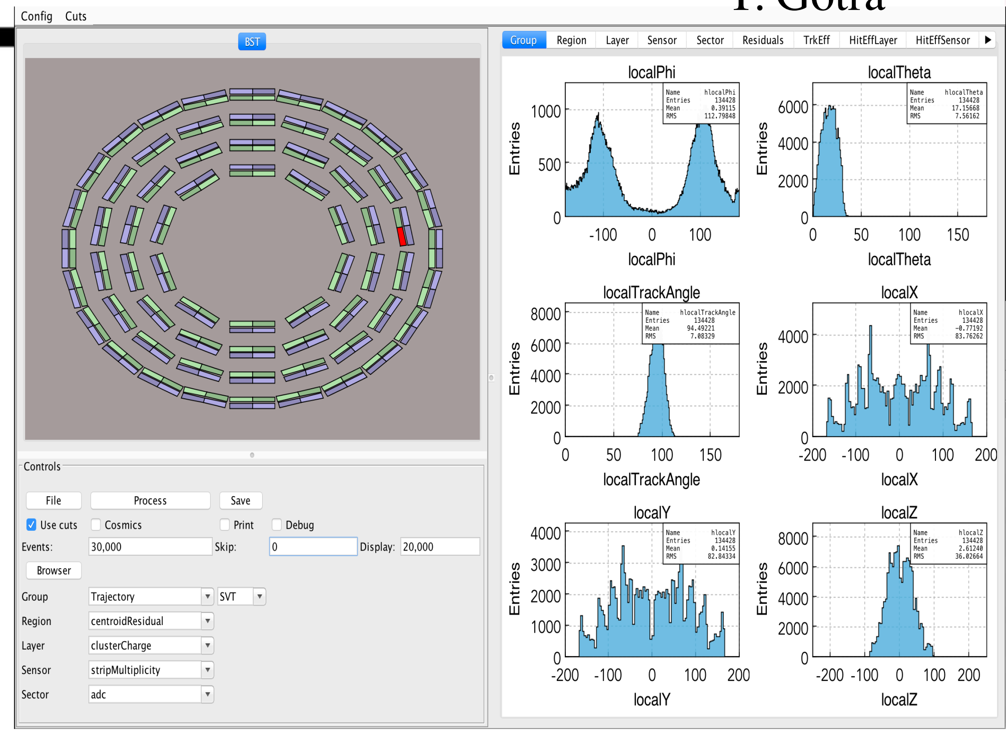
- Histogram selection menus added
- MVT histograms added
- Cut selection menu implemented
- Event skimming added
- Unbiased centroid residuals added
- Efficiencies and resolutions implemented
- Hipo and root output format

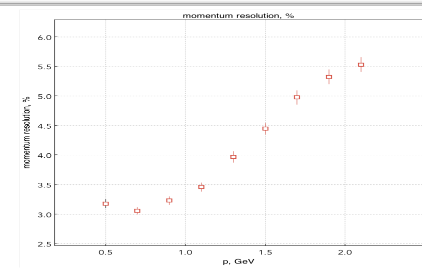
Validations performed

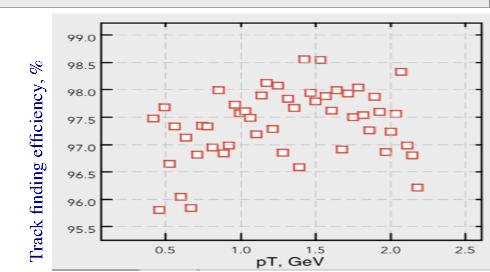
- Reconstruction release validation v0.1 - v0.6
- Single track reconstruction
    - Geantinos, muons, pions
    - Straight (0T) and helical tracks
- Gemc 2.3
- Geometric acceptance
- Discriminator thresholds
- Resolutions (momentum, angular)
- Efficiencies (track finding, hit finding)
- Occupancies

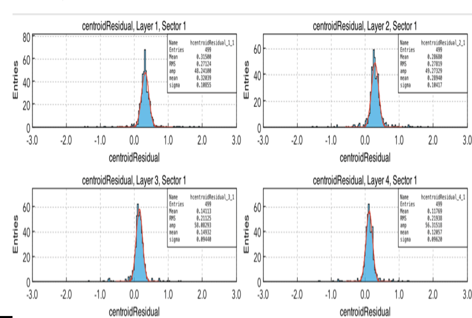
Work in progress

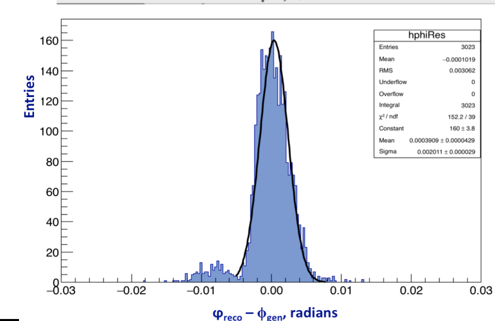
- Misaligned geometry
- Multiple tracks
- Electronic noise
- Local reconstruction
- Lorentz angle
- Documentation











CVT Online Monitoring

Strip Plots/Tracker Maps

2d plot, sensor vs. channel (132x256)

- **channel status (green: good, yellow: masked, red: noisy)**
- **occupancy in percent vs. the strip number**
- **average strip pulse height in ADC counts**
- **width of pulse height distribution in ADC counts**
- **new bad strip (red: strip marked by data quality algorithm but not marked)**
- **chip status map**

Component Plots

Selection of component (sensor) in Detector View, 1D

- **occupancy, vs. the strip number**
- **ADC**
- **BCO**
- **cluster charge**
- **corrected cluster charge (by cos of the track angle)**
- **strip multiplicity**
- **unbiased centroid residual**
- **local track phi**
- **local track theta**
- **local track 3D angle**

Statistics Plots

Mean value and RMS (as error bar) vs. sector, by layer

- **ADC**
- **occupancy**
- **cluster charge**
- **strip multiplicity**
- **unbiased centroid residual**

Summary/Combined Plots

Per layer/region, total

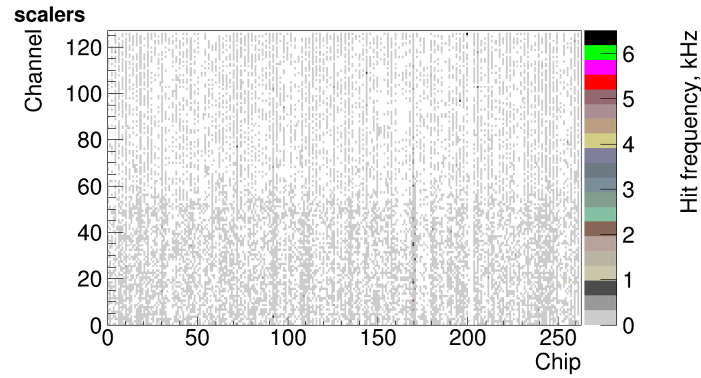
- **hit finding efficiency, occupancy, norm. by nb of strips (event-by event)**
- **ADC**
- **cluster charge**
- **corrected cluster charge (by cos of the track angle)**
- **unbiased centroid residual**
- **strip multiplicity**
- **hit multiplicity**
- **cluster multiplicity**
- **cross multiplicity**

Tracker Object Plots

- **track p, pt, ϕ_0, θ_0, z_0, d_0**
- **track ϕ_0 vs. track θ_0**
- **track normalized χ^2,**
- **track multiplicity**
- **path length**
- **hits per track**

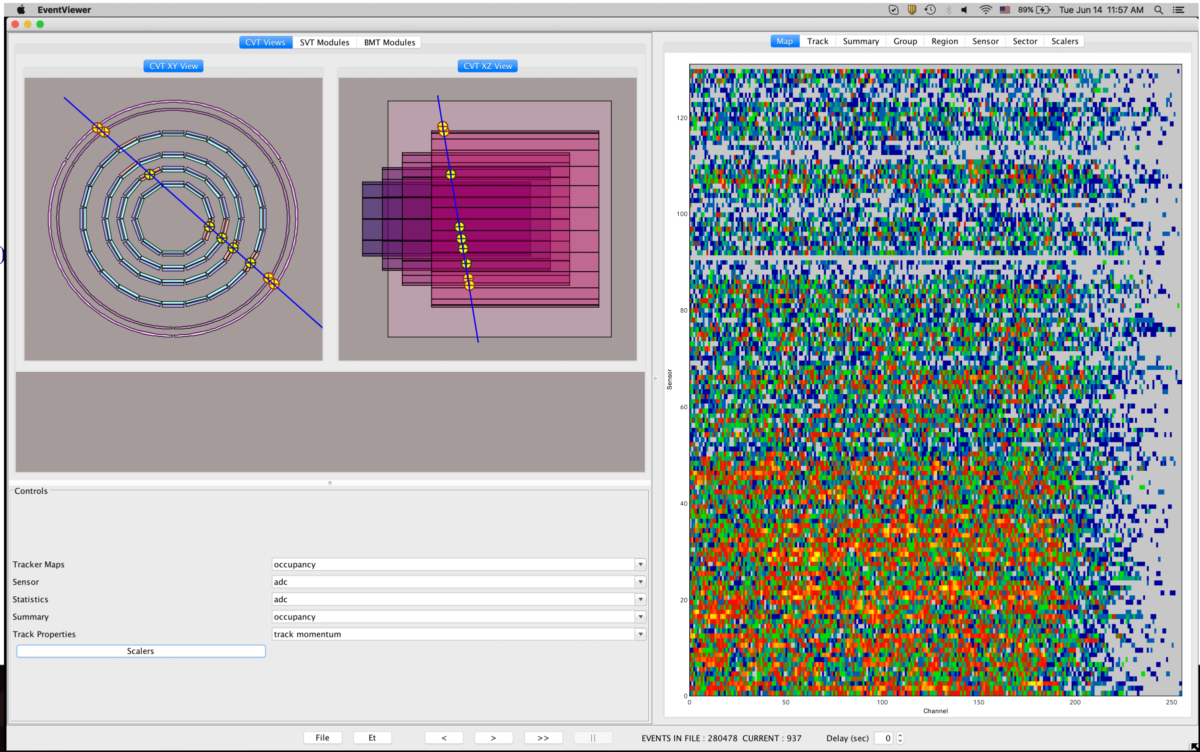
Views:

- **Summary**
- **Report**
- **Shift**
- **Expert**



Monitoring Plots:

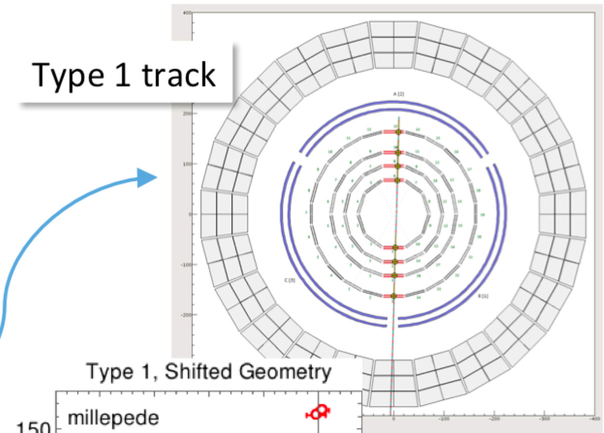
- **long-term (statistics accumulated in the run)**
- **short-term (during the last few minutes or over a few most recent events)**
- **history plots (time history of any quantity with long/short-term plot)**
- **periodic plots (averaged over a fixed number of events)**
- **tracker maps**

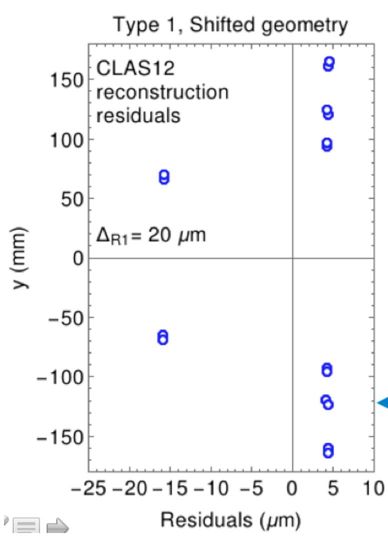


Alignment of the SVT

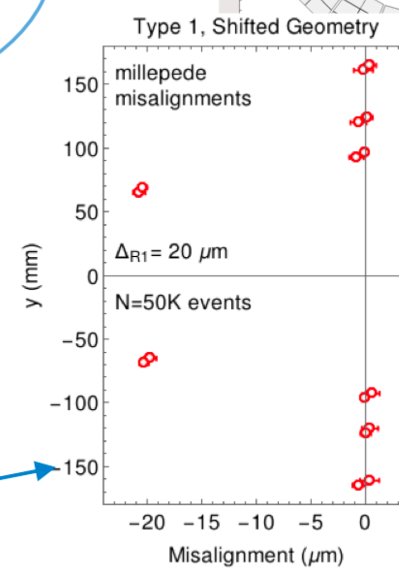
J. Gilfoyle

- Alignment of SVT requires fitting large number of parameters: $N_{regions} \times N_{layers} \times N_{trans} \times N_{rot} = 66 \times 2 \times 3 \times 2 = 792$
- Program **millepede** does linear least squares with many parameters
  - Matrix formulation of least squares method
  - 2 classes of matrix:
    - ➢ Global parameters – the geometry misalignments. Same in all events
    - ➢ Local – individual track fit parameters. Change event-to-event
  - Calculate first partial derivatives of the fit residuals w.r.t. local (i.e. fit) parameters and global parameters (geometry misalignments)
  - Manipulate the linear least squares matrix to isolate the global parameters (geometry) and invert the results to obtain the solution





- Apply to a 'simple' example – Type 1 tracks
  - Use gemc cosmics for testing and validation
  - Initially restrict fit to x direction
  - Use coatjava reconstruction results
  - C++ code to prepare file for fitting
  - Run pede
- Validate by inserting known shift in GEMC
  - Shift layers 1-2 (Region 1) by 20 microns in *x*
  - Shift clearly visible in residuals from reconstruction.
  - Millepede reproduces the shift



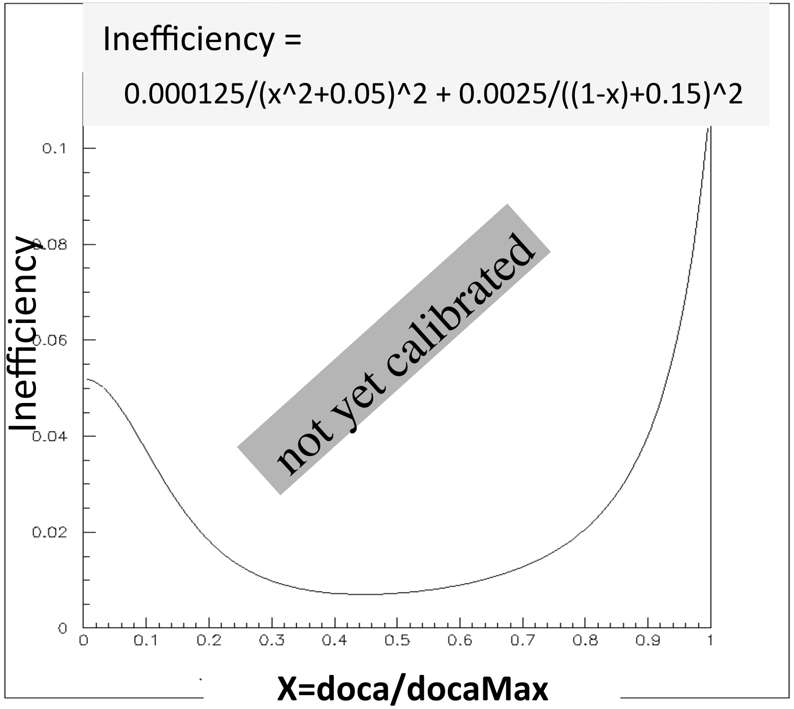
* Geometry implementation in Java framework & validation ongoing (P. Davis [U. Surrey])

DC Reconstruction

- Realistic time smearing and intrinsic inefficiencies in MC
  - using doca RMS in fit
- Time-to-distance parametrization (M. Mestayer & K. Adhikari [U. Miss.])
- Improved noise rejection algorithms
  - secondaries pruner
  - LR ambiguity resolver
- Development of improved Hit-based track parameters (in development)
  - using KF fitting method
  - using segment dictionary & Neural Net (D. Heddle [CNU], M. Catelli [CNU student], L. Lorenti [CNU student])

Simulation of intrinsic wire inefficiencies

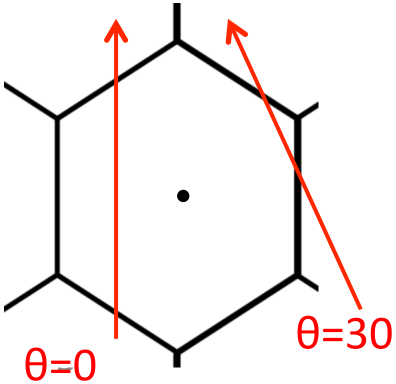
- Three sources of inefficiency:
  - Intrinsic (applies to all wires) – cells don't always fire,
  - Equipment malfunction-related (applies to specifc wires),
  - Background-related (unavoidable knock-on electrons)
- Improved digitization in GEMC
  - parameters added to CCDB: SQlite
  - intrinsic inefficiency (distance dependent) is added in GEMC



The intrinsic inefficiency function:

$$f(X) = scale\left(\frac{P_1}{(X^2+P_2)^2} + \frac{P_3}{((1-X)+P_4)^2}\right)$$

where X = doca/docaMax & docaMax = 2 d_{layer}

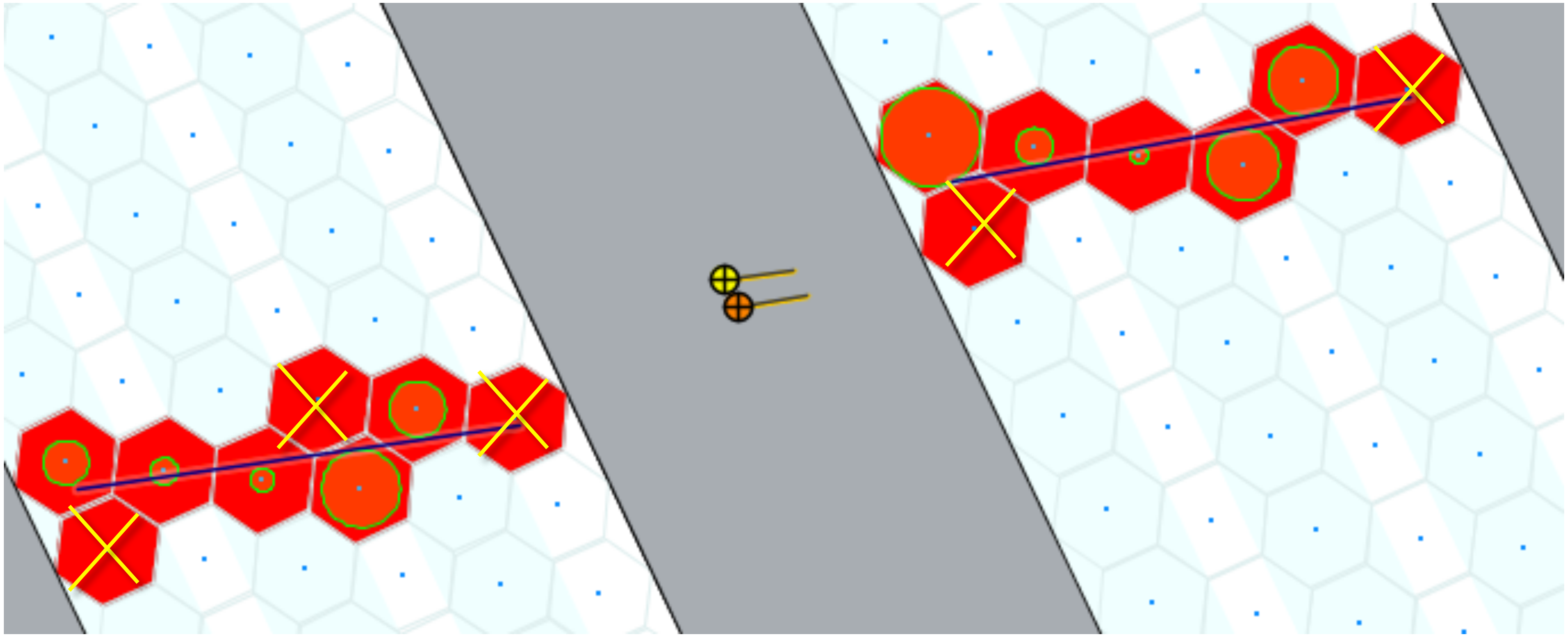


- Hit times generated by GEMC digitization routine will be smeared by a random number with position-dependent magnitudes as given by above intrinsic inefficiency function.
- Same inefficiency function and parameters are used by the track reconstruction software to form error matrix in the Kalman-filter.

Jefferson Lab

Tuning inefficiencies in MC

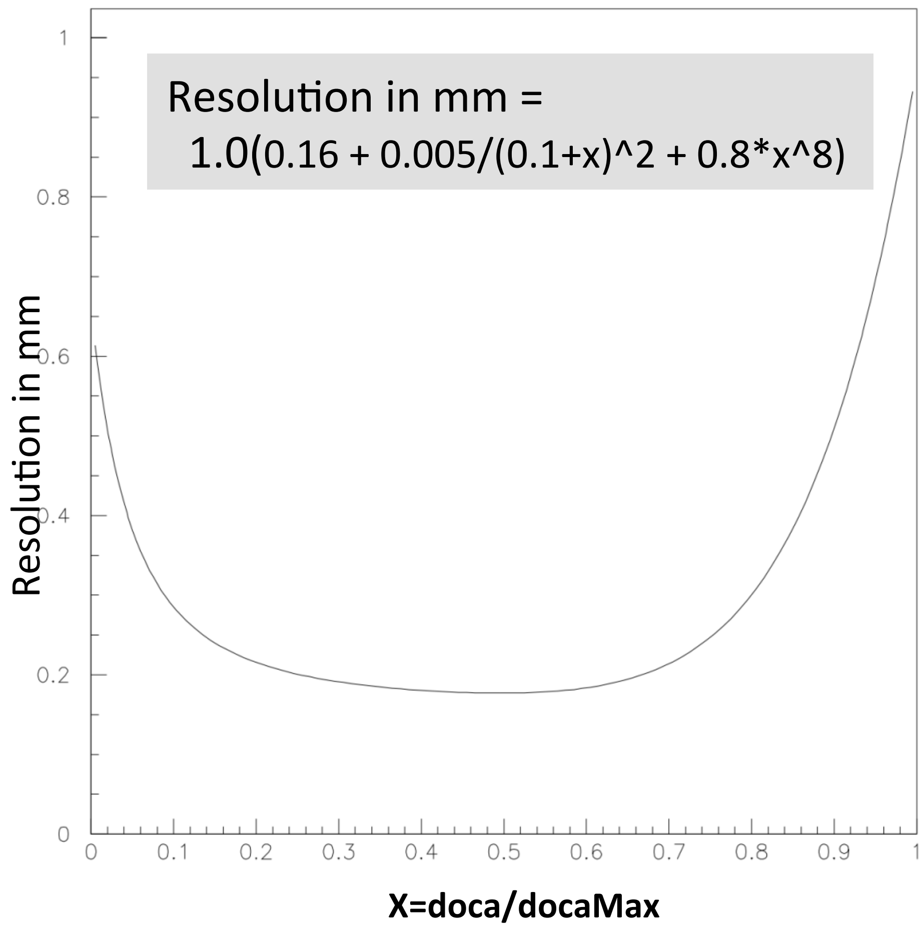
inefficient wire



Simulation of doca resolution

$$scale\left(P_1 + \frac{P_2}{(P_3 + x)^2} + P_4\, x^8\right)$$

- Functional form: M. Mestayer & K. Adhikari
- GEMC implementation: M. Ungaro

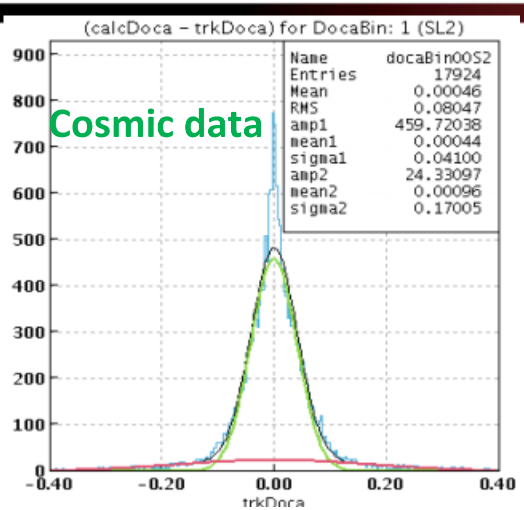


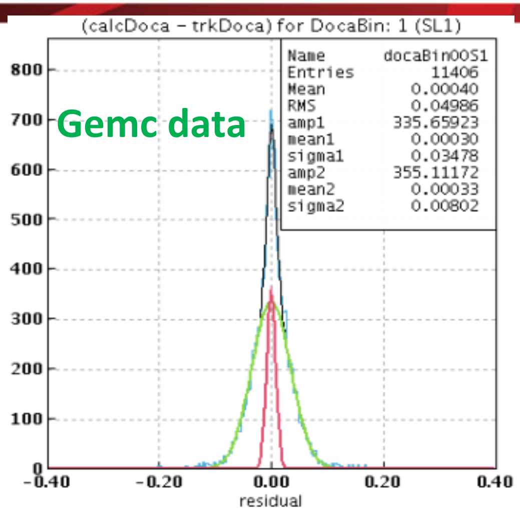
- used to smear docas in GEMC
- used in reconstruction in measurement error in Kalman Gain calculation

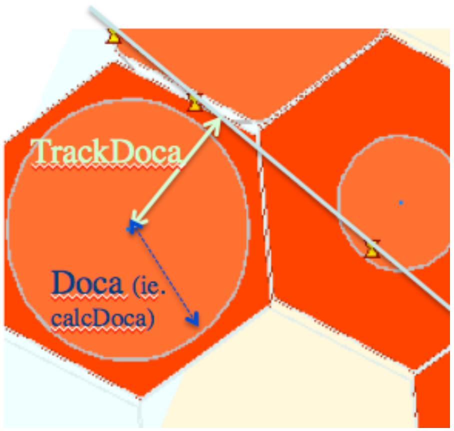
DC-resolution for Cosmics & GEMC

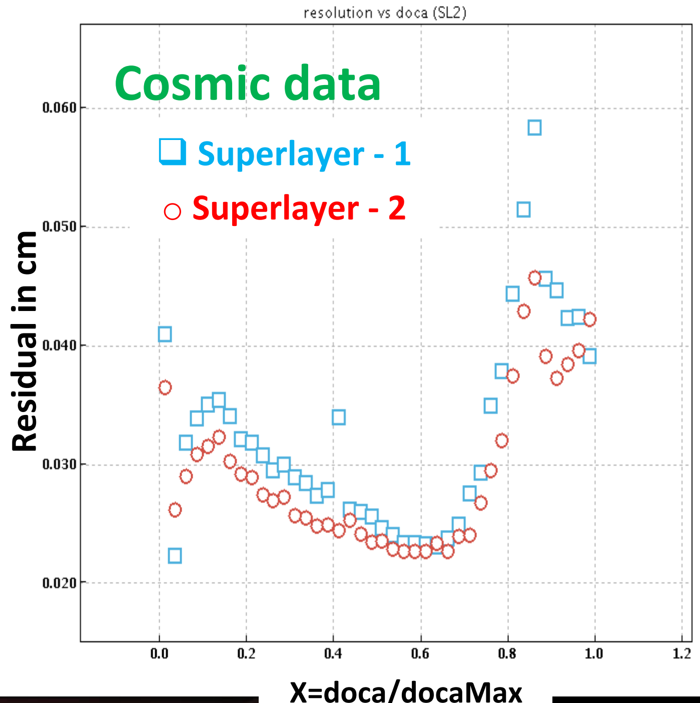
K. Adhikari

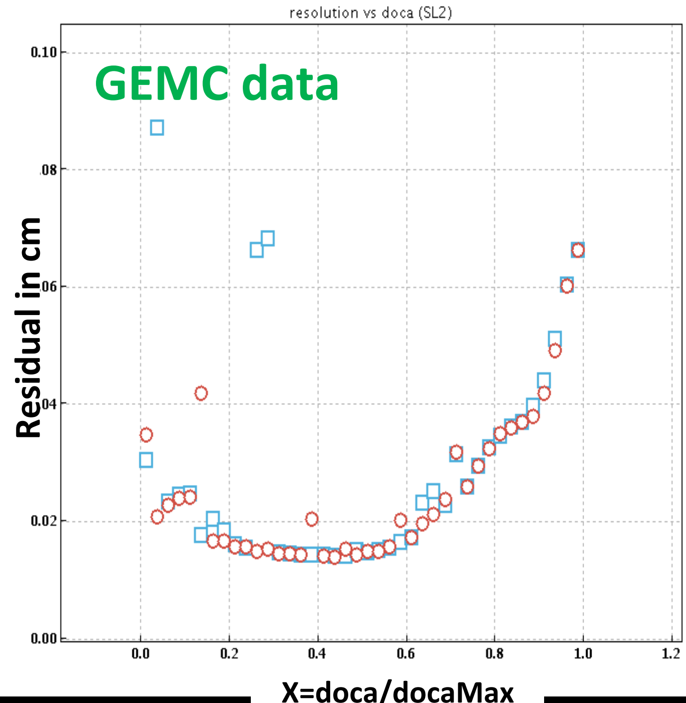
- **Residuals (calcDoca – trkDoca)** in 40 trkDoca bins.
- **Double Gaussian fits** on the residuals
- **Standard deviation** of the central/ narrower Gaussian taken as the **resolution for that bin**.







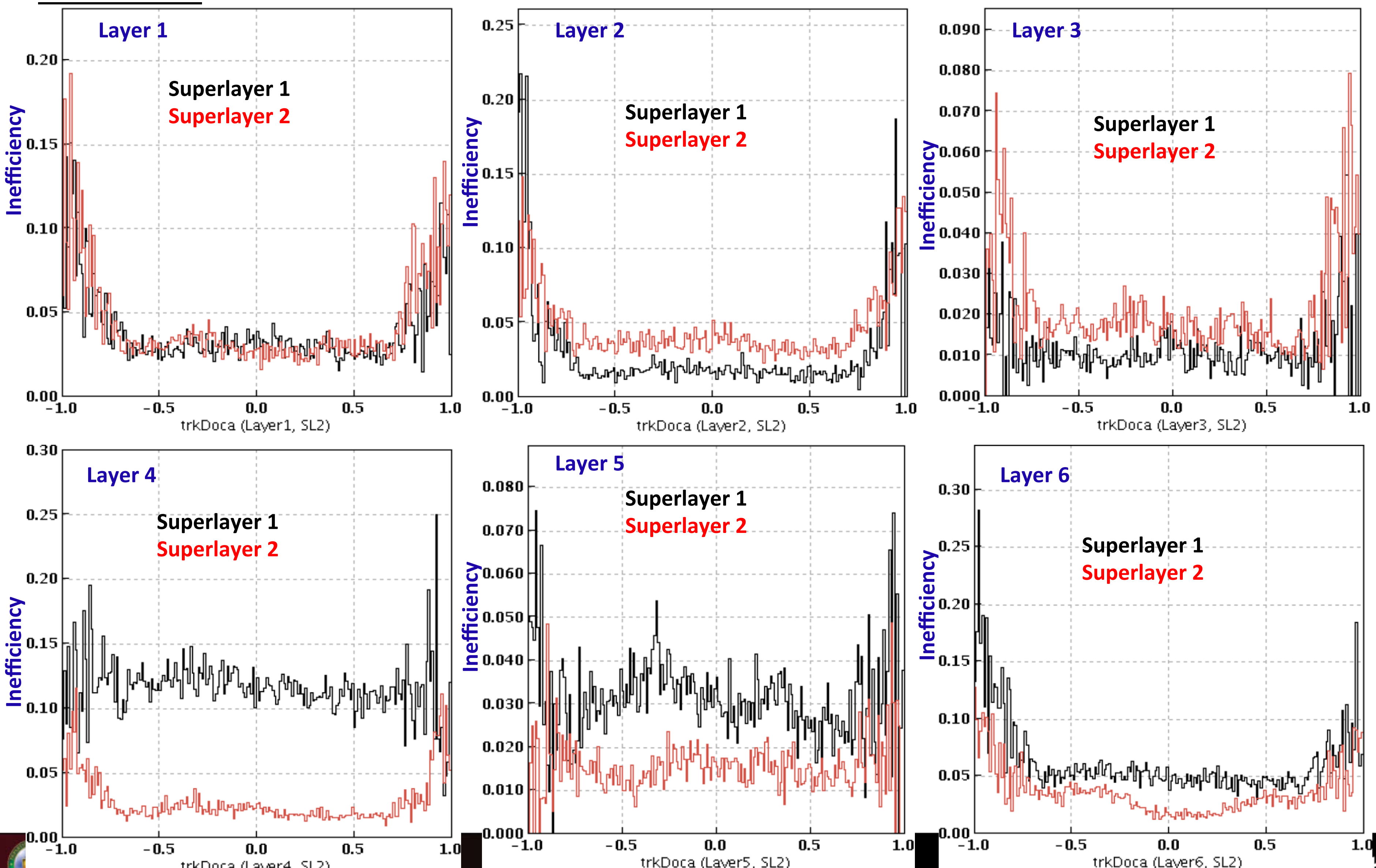




Layer (In)efficiency as function of track DOCA

K. Adhikari

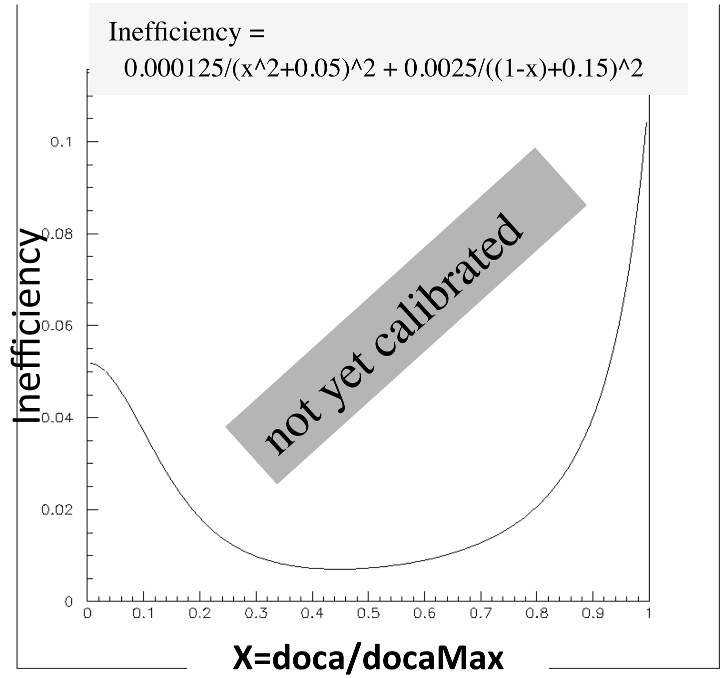
(Cosmic data)

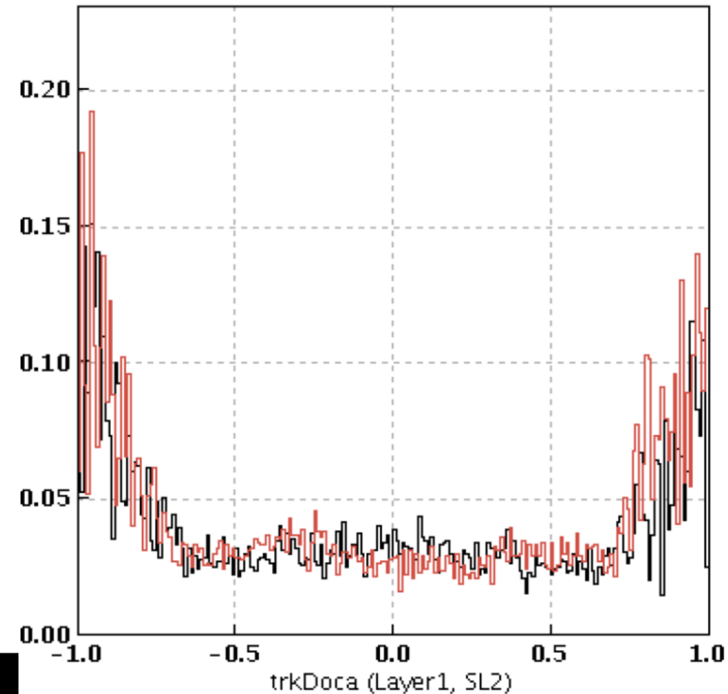


Layer inefficiencies

M. Mestayer & K. Adhikari

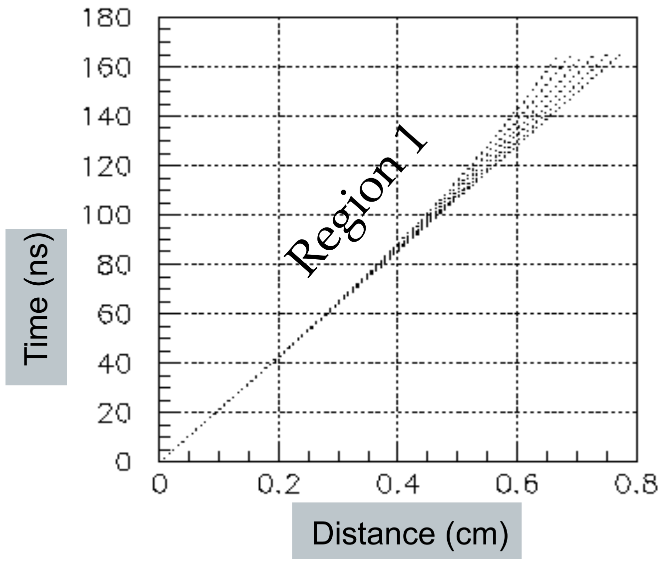
- Studied distance dependence of layer inefficiency for COSMIC data
  - Except of layer 4 in SL1, inefficiency is about 3 to 4 %
  - Layer 4 in SL1 has high inefficiency (about 12%) which seems to be due to voltage issues in some of the channels.
  - Corrections for equipment status (dead channels) not applied yet
  - Time-to-distance function not calibrated yet. (Linear function being used in reconstruction).
- Corresponding study on GEMC data yet to be done.

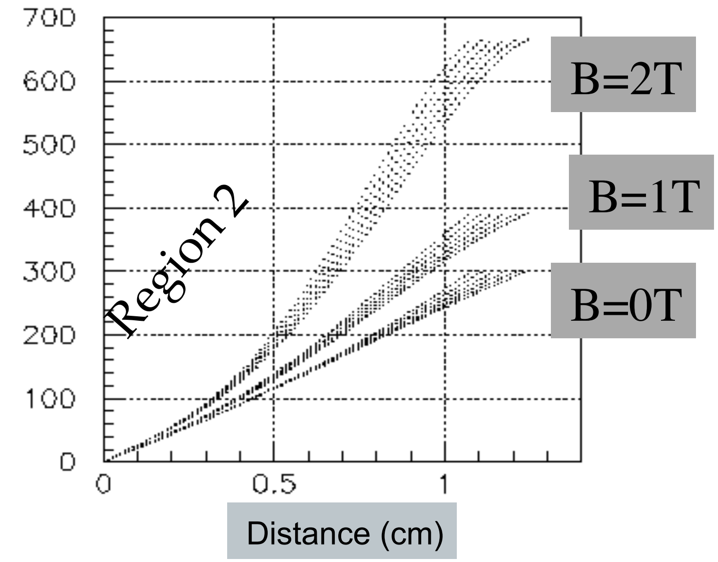


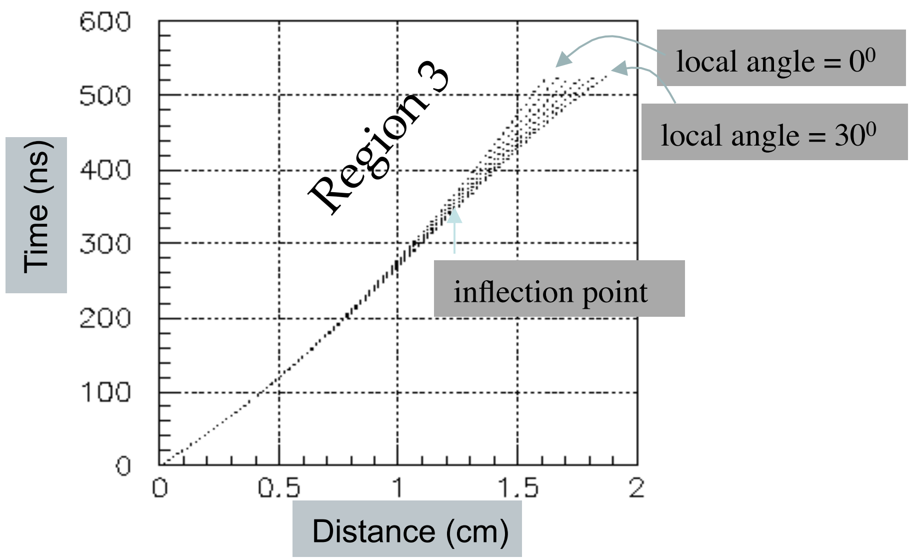


Time-to-distance parameterization

M. Mestayer







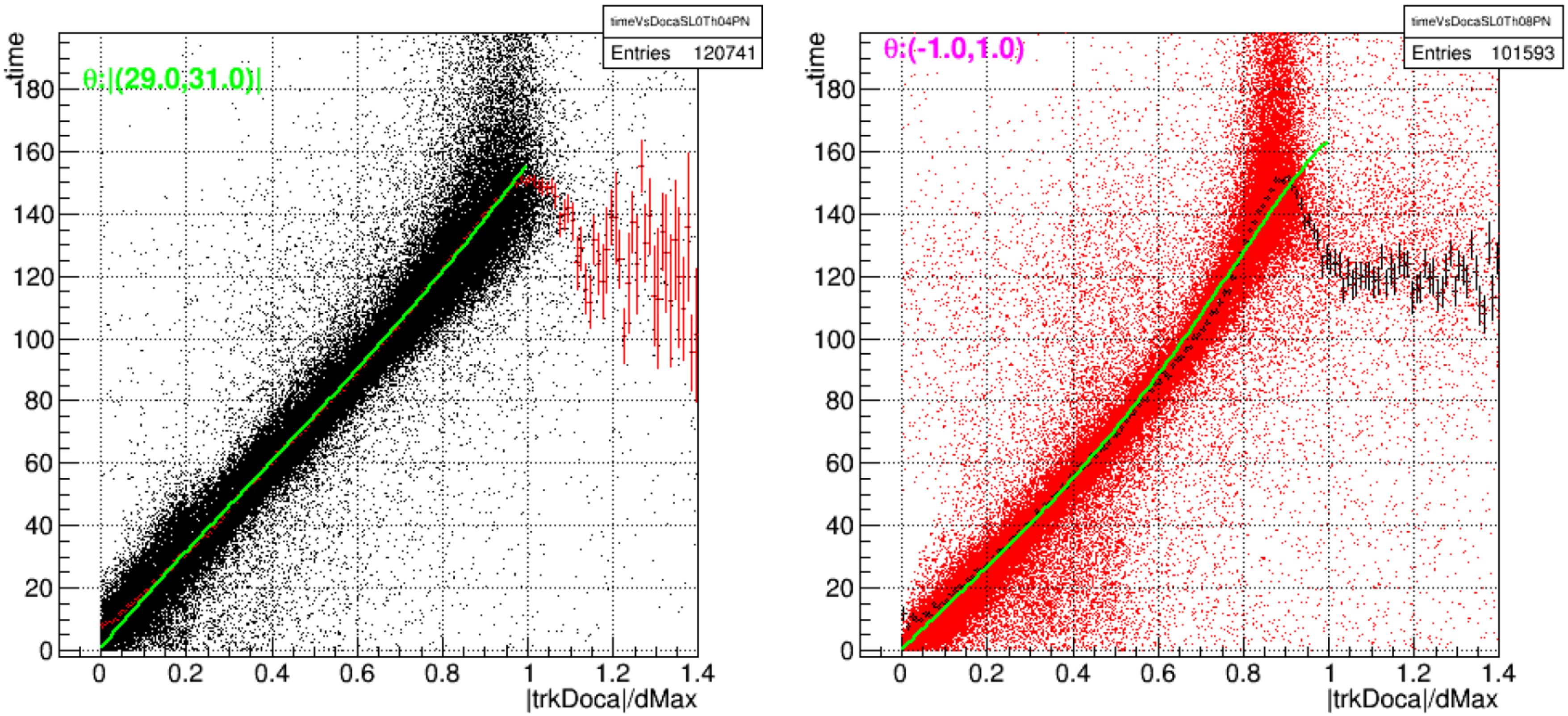
Distance → Time
-local-angle and B-field dependence
-consistent with GARFIELD
-inversion done numerically
-thicker wire → more linear
→ easier to calibrate

Initial parameters & method in software now

Time-to-distance parameterization

Starting equation for 30 degree tracks: K. Adhikari

$$time = \frac{x}{v0} + a\left(\frac{x}{dMax}\right)^{n} + a\left(\frac{x}{dMax}\right)^{m}$$

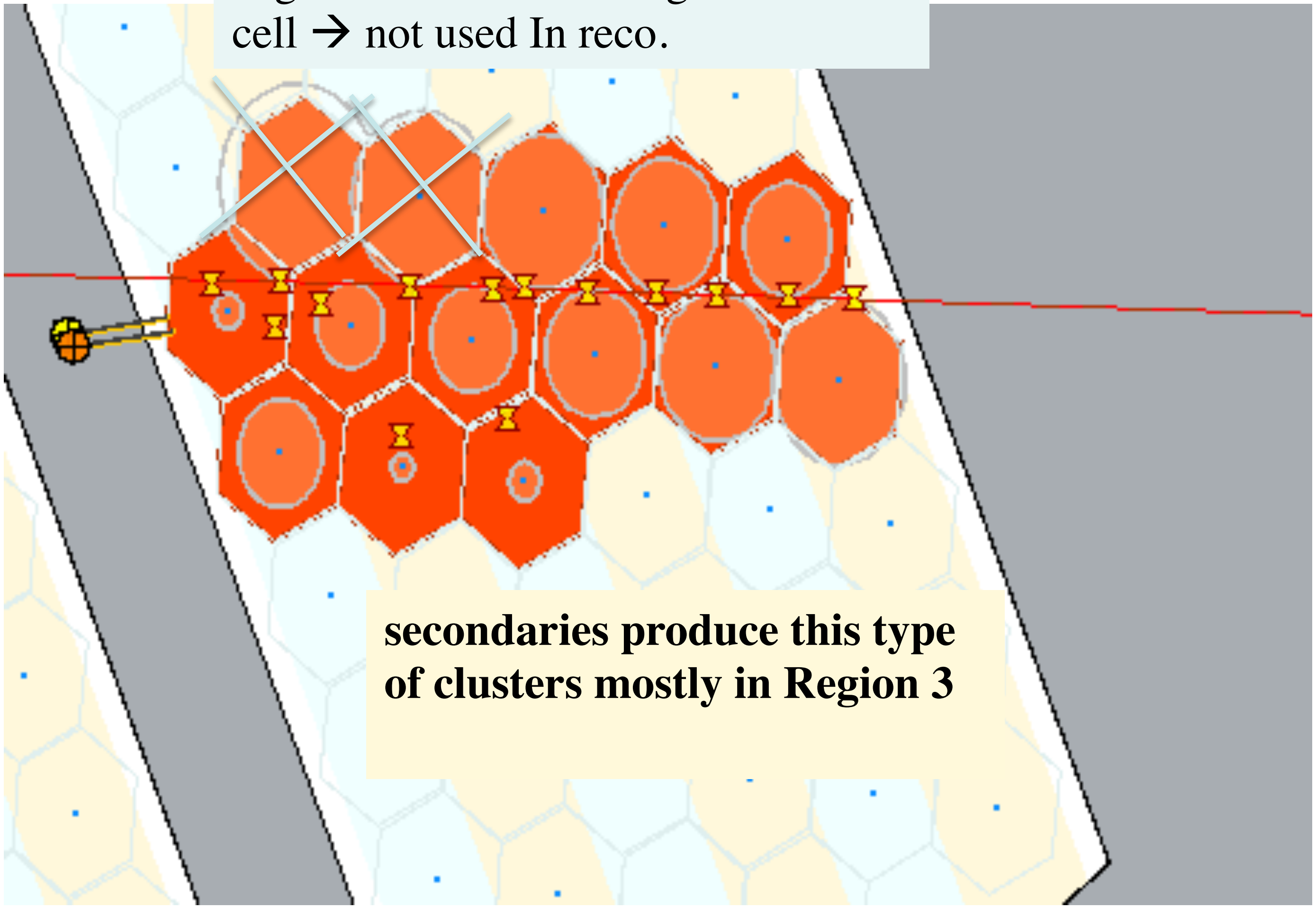


Very preliminary fits on 30 degree & 0 degree tracks respectively

Noise rejection algorithm improvements

Effect of noisy clusters on the reconstruction

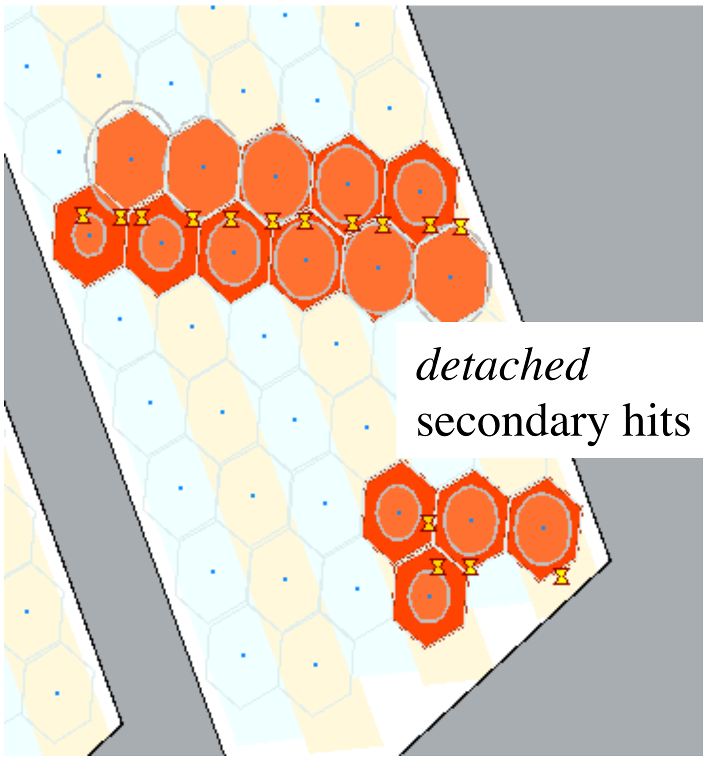
MC sample
4.5 GeV e- @ $\phi = 0^o$, $\theta = 10^o$

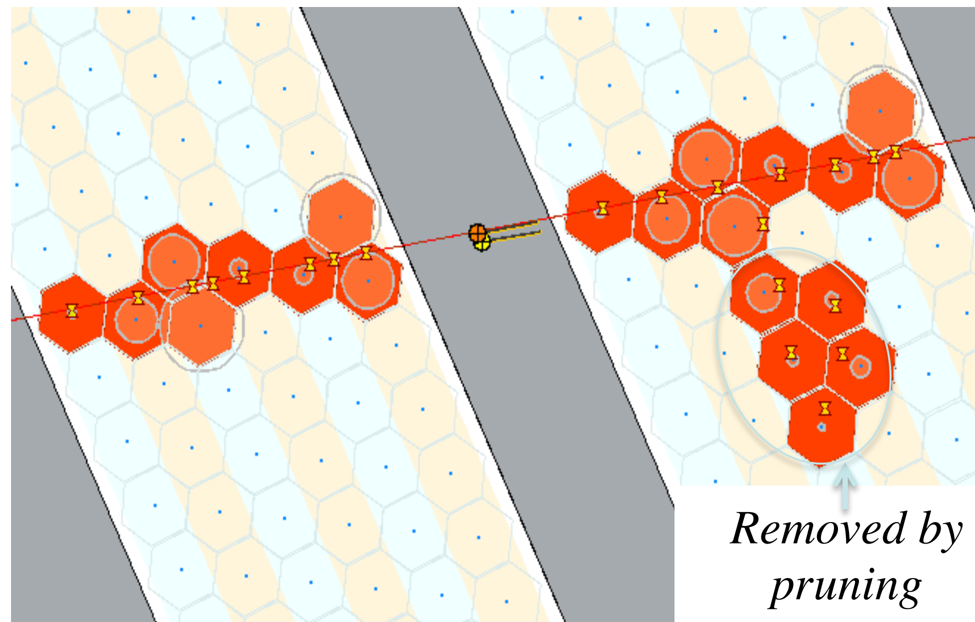


negative times indicating inefficient cell → not used In reco.

secondaries produce this type of clusters mostly in Region 3

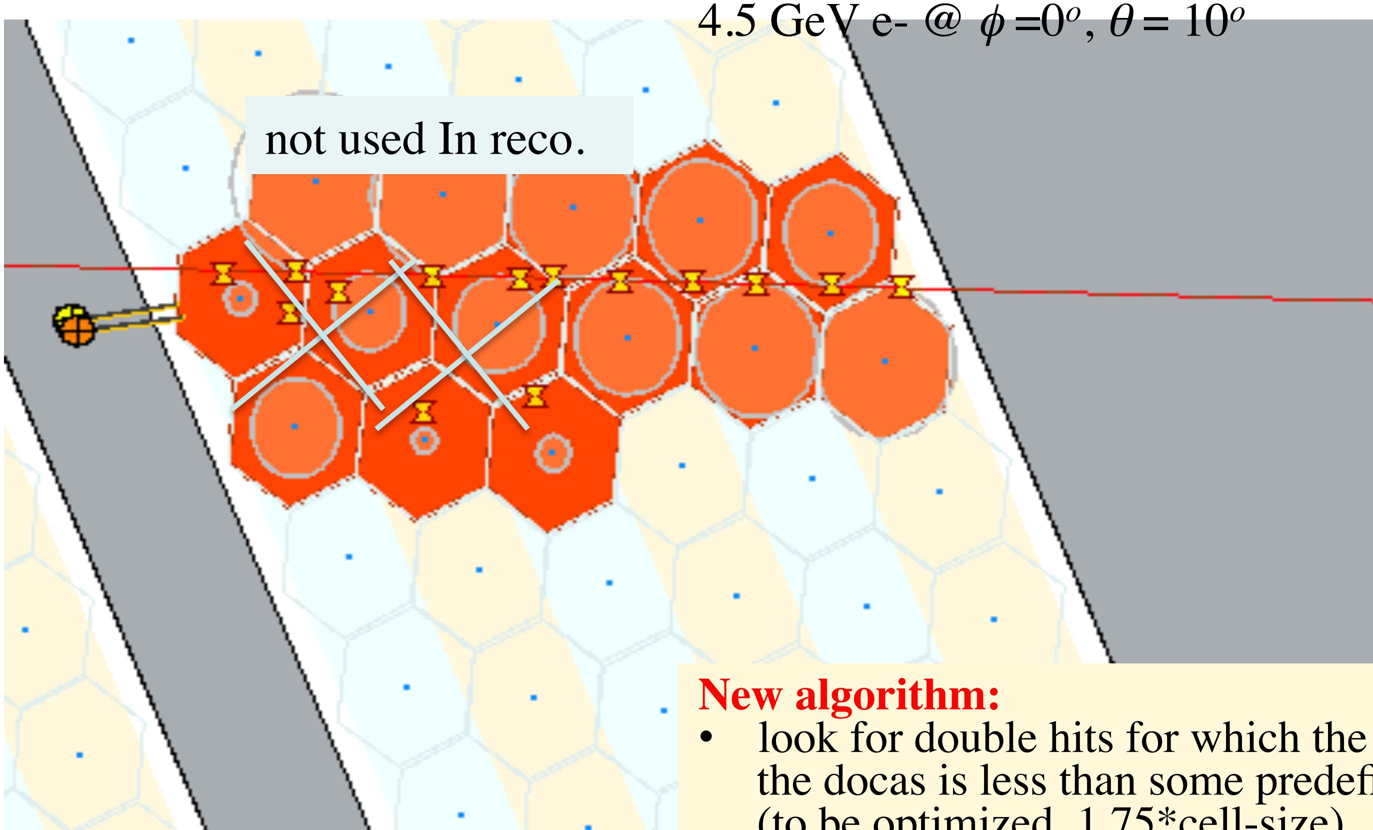
Noisy clusters that do not affect tracking





Noisy Clusters

MC sample
4.5 GeV e- @ $\phi = 0^o$, $\theta = 10^o$

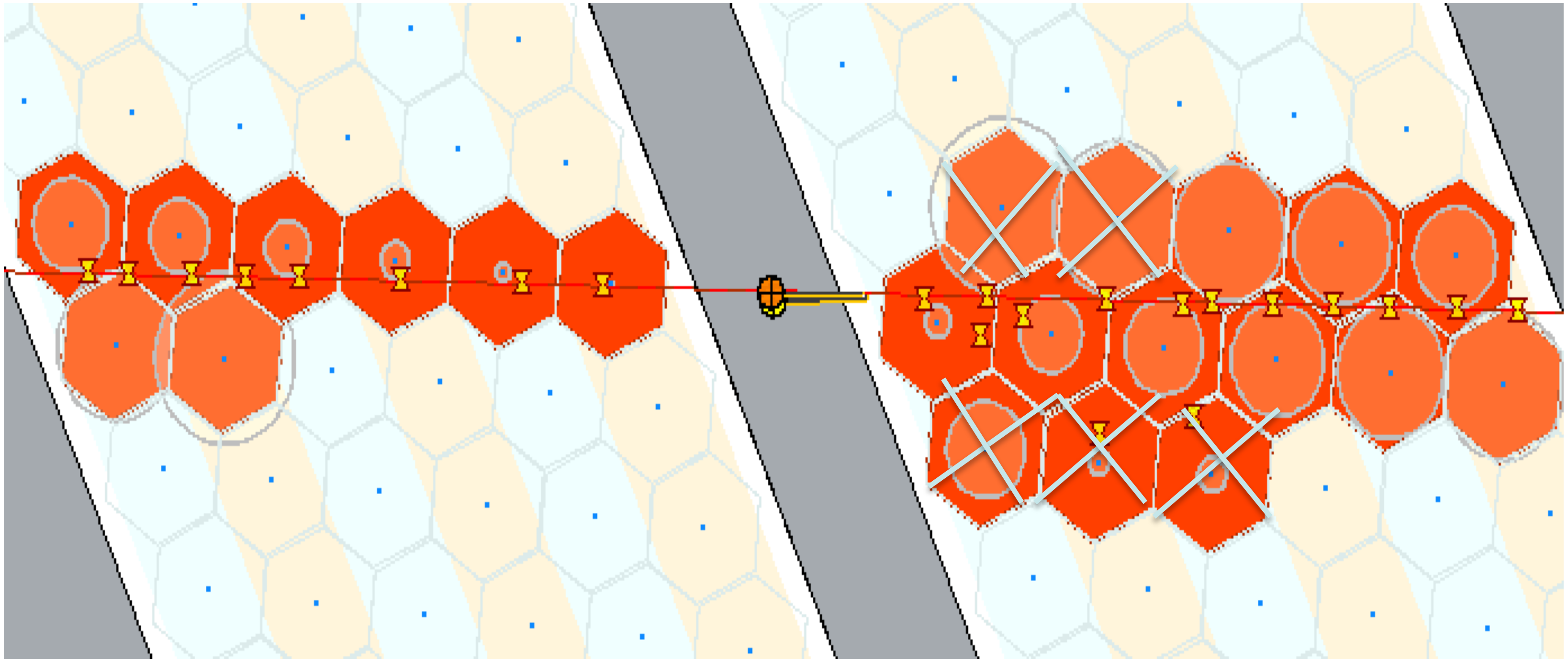


New algorithm:

- look for double hits for which the sum of the docas is less than some predefined cut (to be optimized, 1.75*cell-size)
- refit the cluster for all combinatorials of hits choosing one of the hits in such doublets
- select the best cluster
  - requires to a priori redo hit-based fits as the LR assignment can be wrong

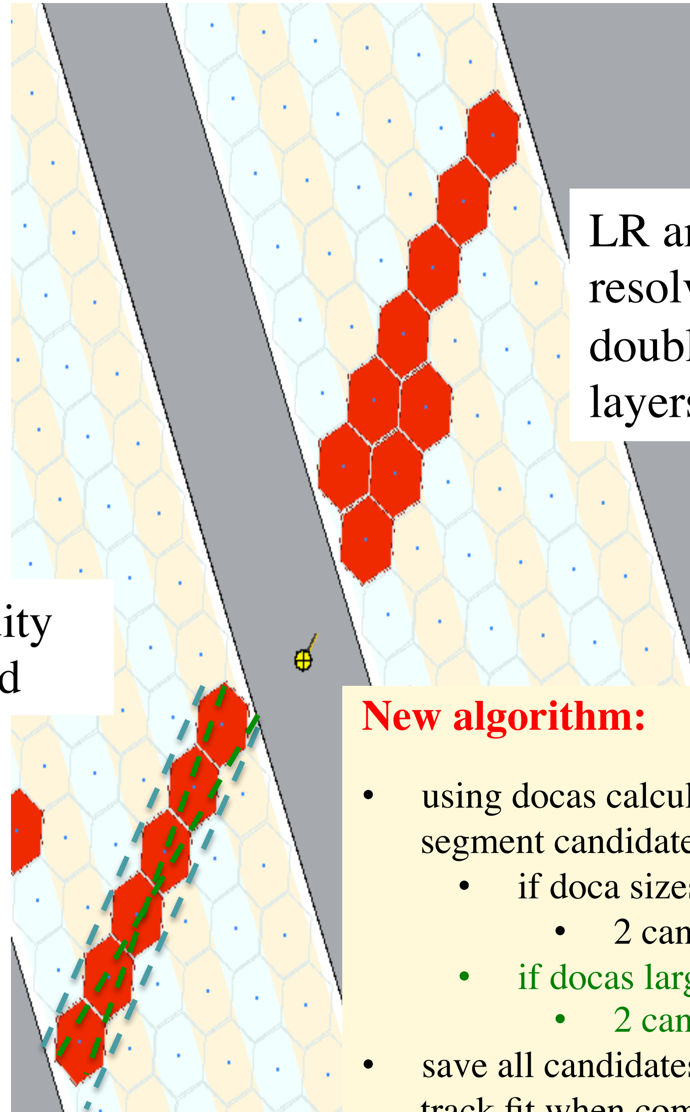
After algorithm implementation

Cross correctly reconstructed



LR ambiguity not resolved for tracks at ~30° in superlayer local coordinate system

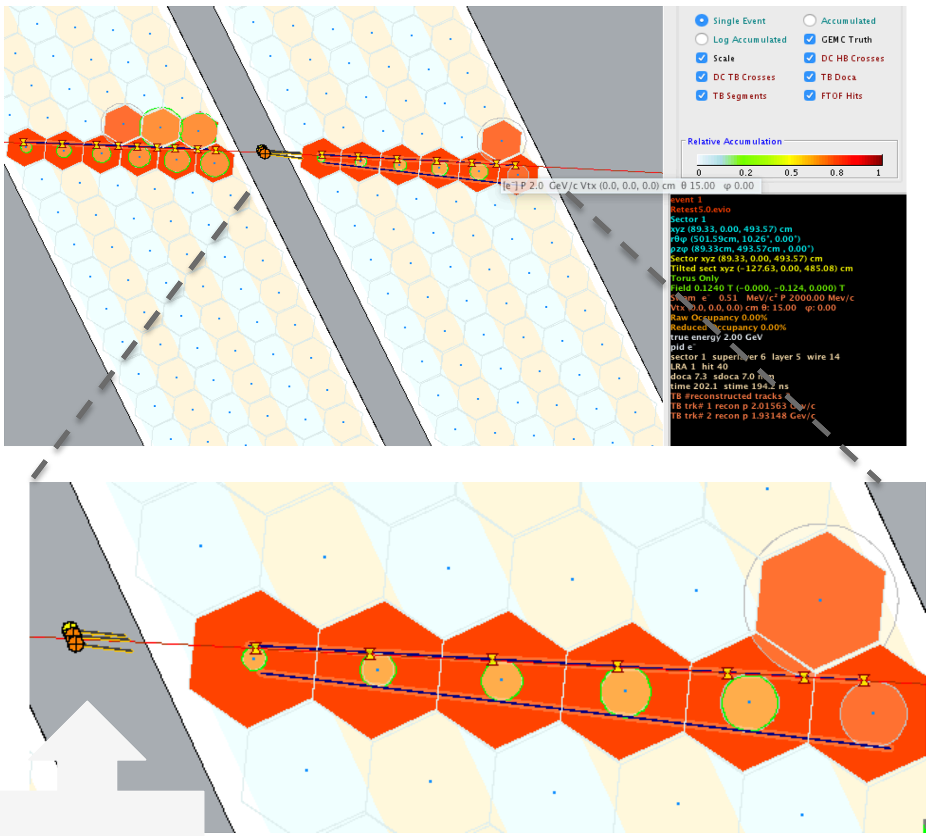
DC cosmic data sample: Region 1 Chamber



New algorithm:

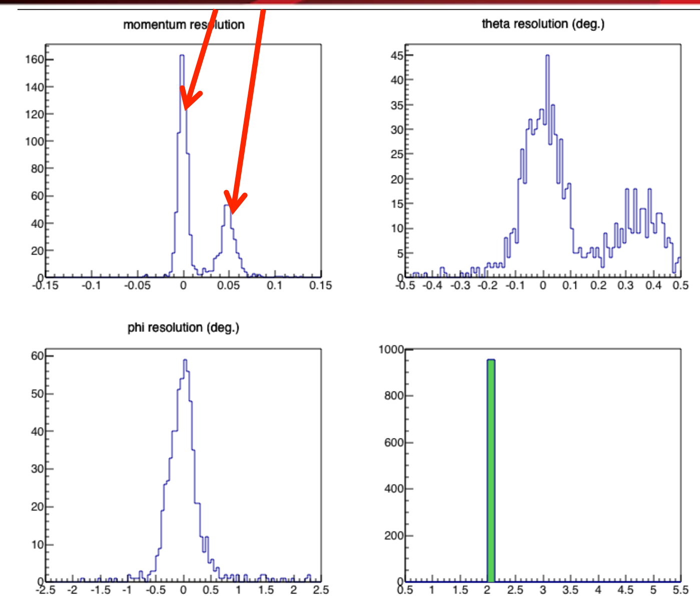
- using docas calculated from times save the following segment candidates:
  - if doca sizes are ~ equal
    - 2 candidates : LR = 1, LR =-1
  - if docas larger at ends of segment
    - 2 candidates
- save all candidates and select the one yielding the best track fit when combined with segments from other regions (for current cosmic sample → save all segment candidates)

Allowing both segments and picking the *correct* one

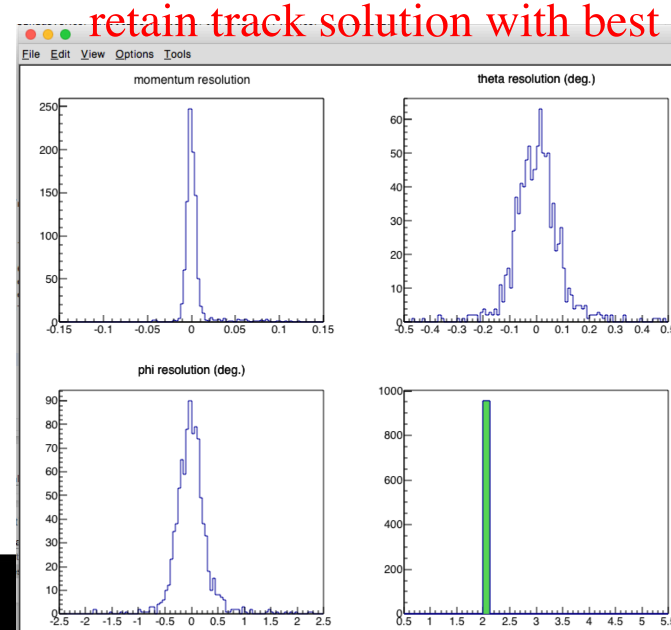


2 crosses → only 1 yields well reco. track

2 track solutions



retain track solution with best chi^2



DC new algorithms and code *restructuring*

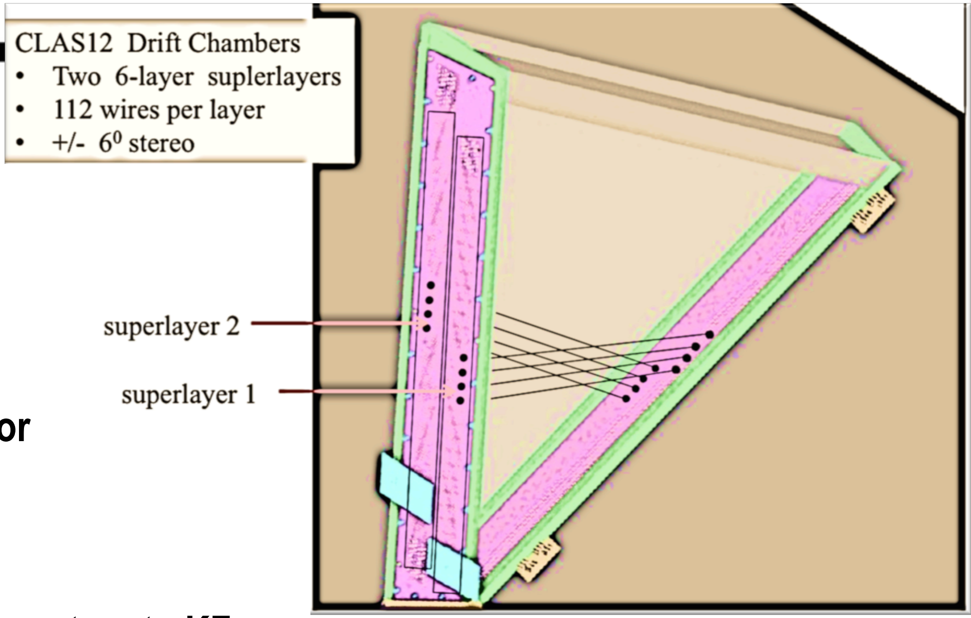
- ClusterCleaner utility class
  - called by ClusterFinder
    - HitBased level
      - hit list pruner
      - find clusters
        - » look for // clusters or X clusters → cluster splitter
    - TimeBased level
      - recompose HitBased Clusters → read from HB bank
      - *secondaries* remover → using sum-docas algorithm
      - LR ambiguity resolver
      - Final fit → cluster line → used in cross calculation
- Status word for cluster:
  - Array: → Can be used in analysis to reject poorly reconstructed segments when high sample purity is required…

layer →	1	2	3	4	5	6
nb hits in layer	0,1,2	0,1,2	0,1,2	0,1,2	0,1,2	0,1,2
LR ambiguity sum	-1,0,1	-1,0,1	-1,0,1	-1,0,1	-1,0,1	-1,0,1
Ave. nb hits passing residual cut (350 μ)	0,1,2	0,1,2	0,1,2	0,1,2	0,1,2	0,1,2

Hit-based Tracking Improvements

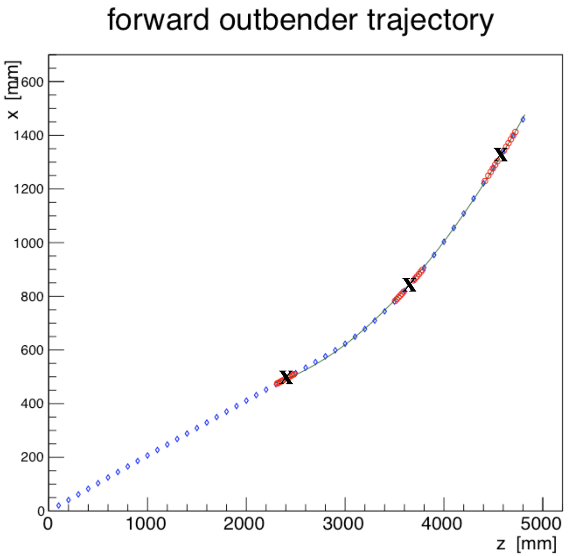
- Previously hit-based tracking used only to select a track candidate.
- Very rough estimate of track parameters using a simple approximation (next slide) → very poor momentum resolution
- Redesign code to improve hit-based track parameters estimates to use them to match track to outer detectors and get start time

DC Reconstruction Algorithms (reminder)

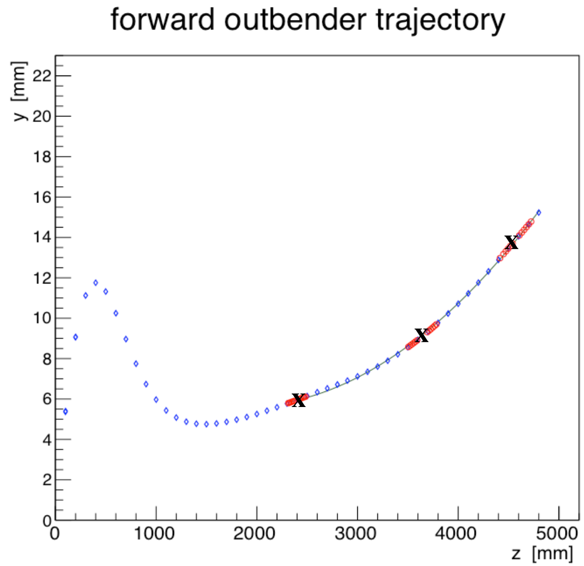


- Obtain a trajectory from hit-based track segment reconstruction
  - Fits to the wires → extended to a plane → point & direction
- **Gives a "cross" object a position and direction vector**
- **Add raw timing information to refine the hit position**
- **Fit to the crosses to obtain a trajectory → Initial parameters to KF**

In x – z plane



In y – z plane



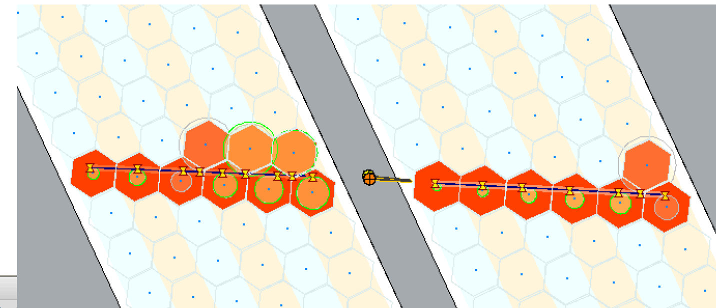
$$\frac{q}{p} = \frac{theta3 - theta1}{0.3 \int B\,dl}$$

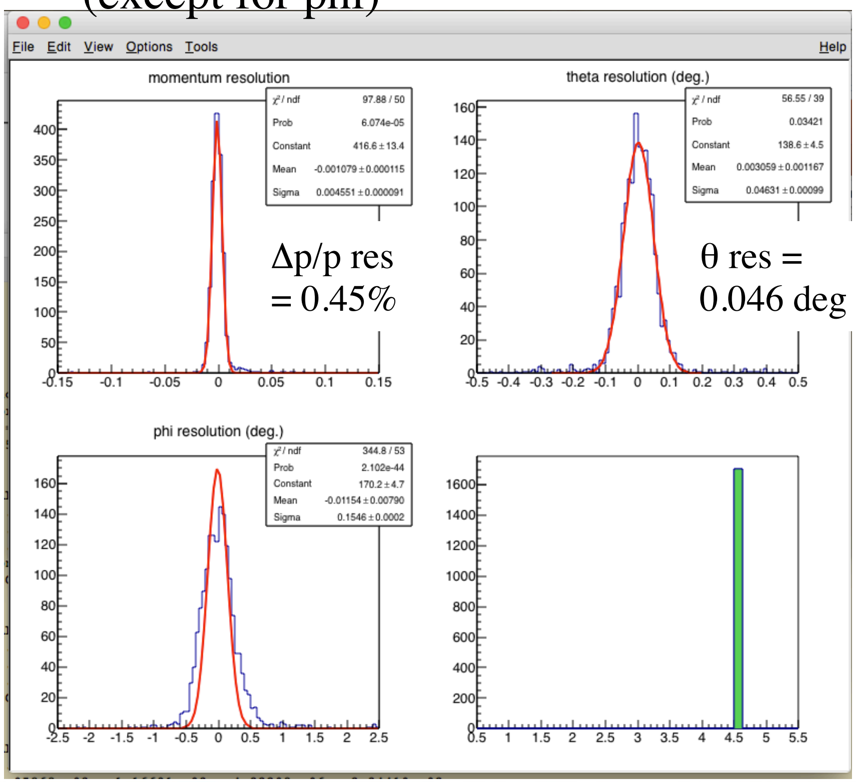
← Quadratic fit

Test of implementation

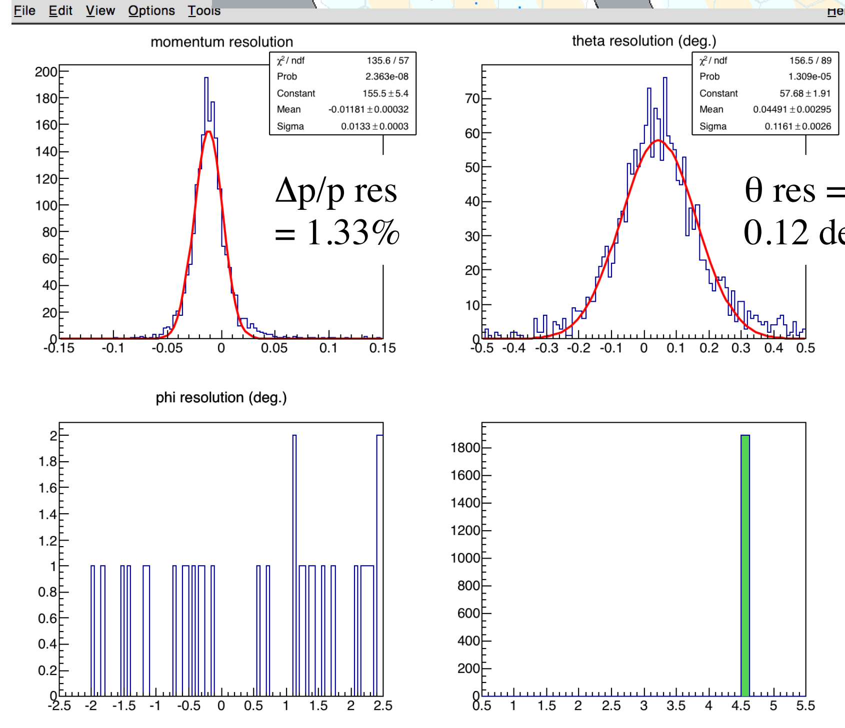
2 GeV e- @ 15 deg θ, at midplane

- Set times to zero (i.e. hit-based) and run KF using wire positions & hit uncertainties of cell-size/sqrt(12) ➔ did not work
- Set times to zero and run KF using segment fit values at measurement plane (fixed z) → *kinda* worked (except for phi)





Time-based



Hit-based

Tracking Timeline

Central Tracking:

- tracker alignment code ready — 4th quarter 2016
- SVT + BMT (4+1) code optimization (unbiased residuals, angular resolution improvements) — 3rd quarter 2016
- SVT + BMT (3+3) configuration implementation (geometry & reconstruction) — 2nd quarter 2017

Forward Tracking :

- Time-to-distance calibration & implementation in reconstruction — 4th quarter 2016
- Hit-based tracking parameters improvements (needed for Event Builder) — 3rd quarter 2016
- use of all calibration constants and status tables in reconstruction — 4th quarter 2016
- integration with FMT in reconstruction — 3rd quarter 2016
- alignment code and magnet mapping ready — 1st quarter 2017

FW Trkg & PID:

- FT-Trk java code ready and integrated with FT system — 4th quarter 2016

Event Builder:

- event reconstruction chain ready (e- or hadron id, start time from hit-based tracking, detector matching, full PID using all available detector responses) — 3rd quarter 2016