Update on SoLID Track Reconstruction

Ole Hansen

Jefferson Lab

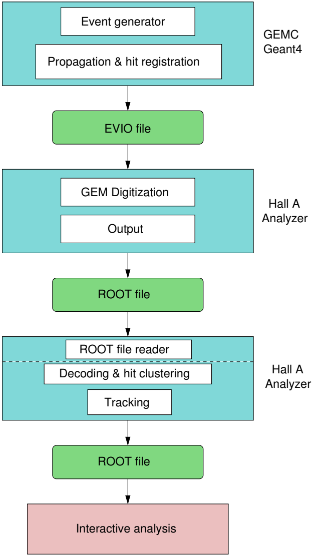
SoLID Collaboration Meeting
March 22, 2013

TreeSearch Reconstruction Algorithm

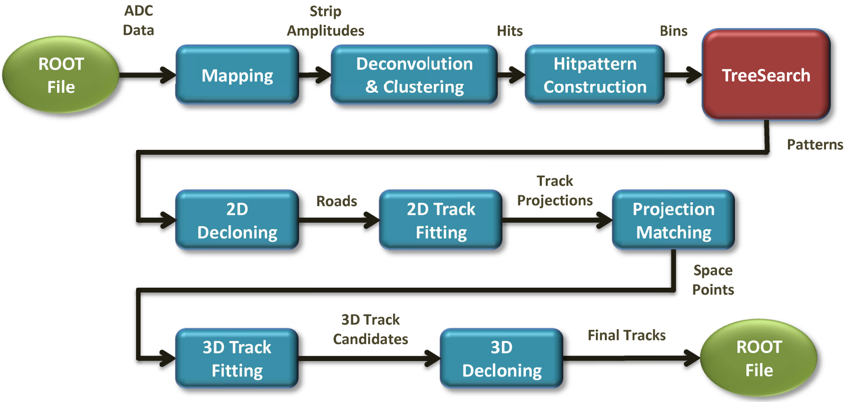
- Global recursive template matching
- Pros
  - Efficient. High speed: $\mathcal{O}(\log N)$. Small memory footprint: $\mathcal{O}(10\text{ MB})$
  - No seed point needed
  - Available in Hall A analyzer
  - Successfully used with BigBite data and SBS simulations
- Cons
  - May not fully solve the problem: requires (nearly) straight tracks
  - Allowing for small track curvature adds complexity
  - Code must be adapted to SoLID geometry

Track Reconstruction Simulation

- `solgemc` EVIO files as digitization input (S. Riordan)
- GEM digitization based on SBS work (E. Cisbani, R. Holmes)
  - APV25 pulse shape simulated
  - Ad-hoc noise simulation (random time offset)
  - No other detectors digitized yet
  - Partial passthrough of generated data (tracks, vertices)
- ROOT file interface
- Tracking



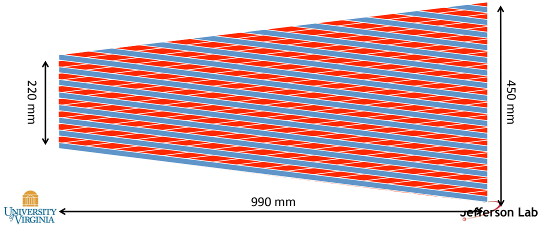
TreeSearch Track Reconstruction Chain (GEM version)



Code & Algorithm Modifications Made for SoLID

- Support SoLID geometry
  - Decoder for simulation output
  - Support detector positioning in cylindrical coordinates
  - Cut on non-rectangular active detector area
  - Particular difficulty: Chambers may have angular an offset!
- Make all sectors appear as one spectrometer, not 30 separate ones
  - Automatically supported in C++ analyzer, but could be more efficient
- Note yet done: Allow for (small) track curvature in 2D and 3D fits
  - Need efficient algorithm
  - Implement parameter range limits
  - Stability?

GEM Chamber Strip Layout (illustration from Nilanga)



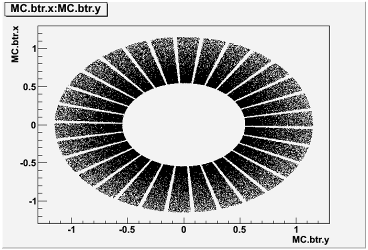
- Strips in different planes MUST be parallel for TreeSearch.
- If chambers in different planes have angular offsets, then strips must be rotated wrt chamber frame in the offset planes. (Sorry, no picture.)
- Probably don't want to manufacture GEM chambers with rotated strips!
- If chambers are to have angular offsets, and GEM chambers are to have strips as shown above (not rotated), then the tracking algorithm must be able to handle non-parallel strips in different planes. Not impossible, but harder.

SoLID Track Reconstruction: 1st attempt

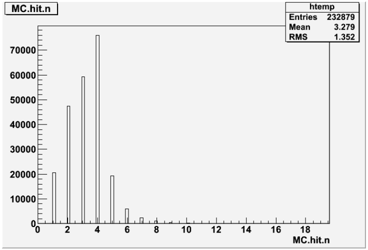
- "Ultra-clean" input data
  - Muons, no field
  - Electrons, with field (not yet analyzed)
  - Very limited materials (basically only the trackers)
  - No background from target
- Full reconstruction chain
- Standard cuts
  - Require 3/4 hits per coordinate
  - Allow 1 missing amplitude correlation
  - Accept wide χ^2 range for fits (up to about 10/dof)

MC input data: tracks, hits (All plots are for "muons, no field")

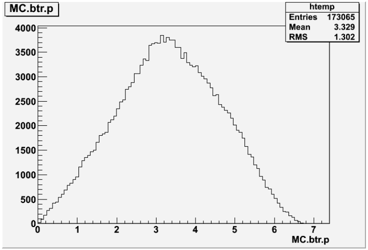
Hit position at first GEM plane



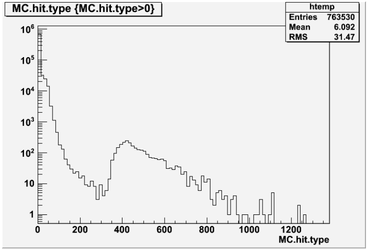
Number of hits per event, expect ≈ 4 (= no. of planes)



Momentum of interacting particle

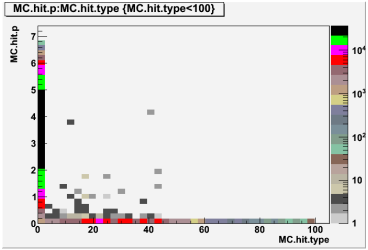


Hit "type", > 1 are secondaries

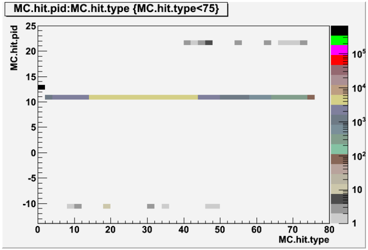


MC Secondaries

Momentum vs type → secondaries have very low p

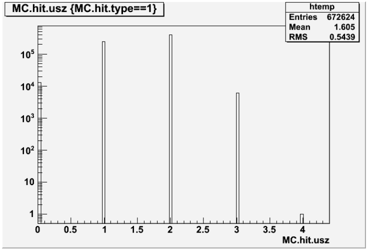


PID vs type → secondaries are mostly e^{-}

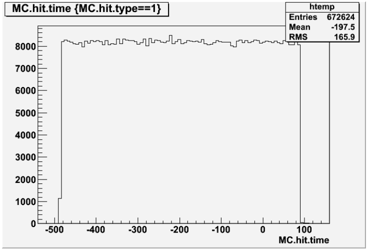


Digitization

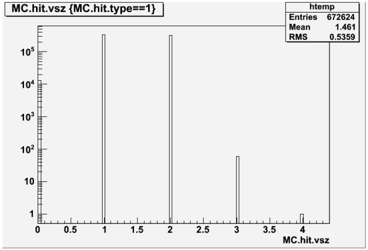
Cluster size, *u*-coordinate



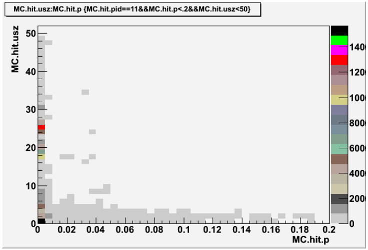
Digitization time offset (BUG!)



Cluster size, *v*-coordinate ($\neq u \longrightarrow$ BUG?)

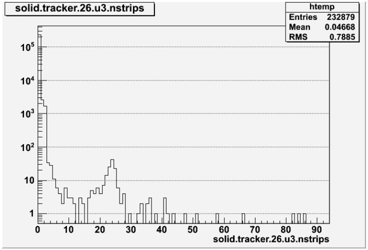


Cluster size vs momentum: low p e^- $\longrightarrow$ huge clusters

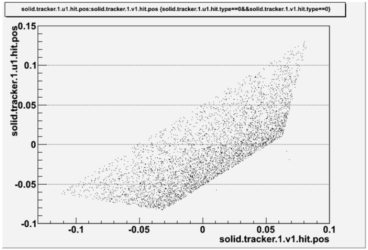


Decoding & Clustering

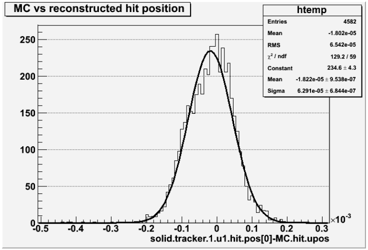
Number of active strips → low occupancy



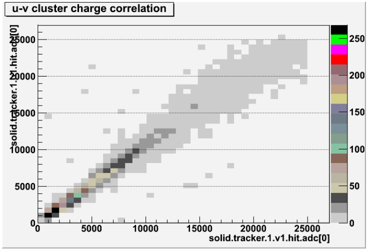
Reconstructed hit coordinates u vs v



Reconstructed hit position accuracy ≈ 70 μm

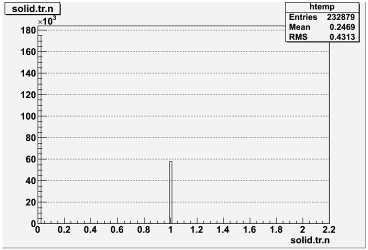


2D readout cluster charge correlation

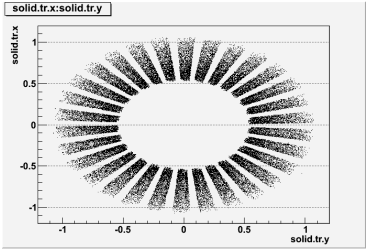


Track Reconstruction

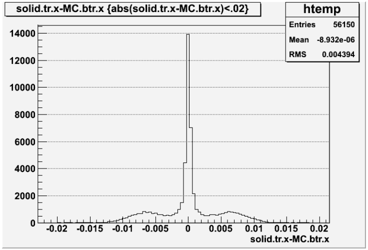
Number of tracks found: ≈ 35% efficiency (173k MC tracks)



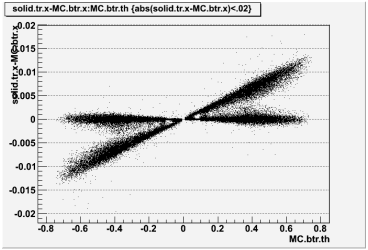
Reconstructed track coordinates at first GEM plane



Track x-coordinate residual → BUG?



x residual vs track $\theta_{TRANSPORT}$ → BUG!



Observations

- It works!
- Digitization still has problems
  - Time offset for trigger tracks
  - Small cluster size
- Fairly low tracking efficiency, but not surprising given still un-optimized items:
  - Digitization
  - Detector and GEM strip alignment
  - Reconstruction parameter tuning (many available)
- Track residuals look really "interesting". Bug? Alignment problems?

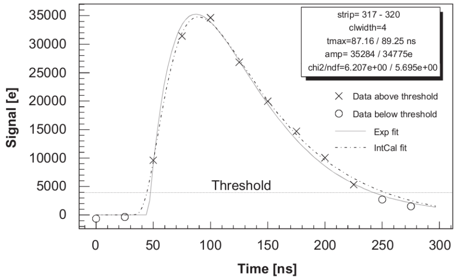
Next Steps

- Address obvious problems from previous slide. Should really get close to 100% tracking efficiency.
- Analyze "electrons with field" to study effect of track curvature
- Simulate realistic conditions
  - Add all materials
  - Add background
  - Add vertex reconstruction
- With full realistic simulation, get estimates for
  - Tracking efficiency
  - Vertex resolutions
  - Ghost & clone track rate
  - Computing performance

Backup Slides

APV25 Pulse Shape Deconvolution & Noise Filtering

S. Gadomski *et al.*, NIM A 320, 217 (1992)



- For first-order RC circuit, signal amplitudes s_k can be deconvoluted using three measured values v_k:

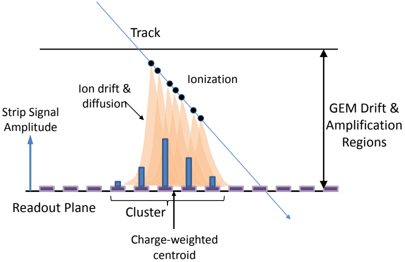
$$s_k = w_1 v_k + w_2 v_{k-1} + w_3 v_{k-2}$$

$$w_1 = e^{x-1}/x, w_2 = -2e^{-1}/x, w_3 = e^{-x-1}/x, \text{ where } x = \Delta t/T_p$$

$$A \approx \sum_{k=1}^{3} s_k$$

- Reject noise by cutting on ratios, $r_1 = v_3/v_1$ and $r_2 = v_2/v_1$, requiring rising slope

GEM Hit Clustering



- Signals on adjacent readout strips typically belong to a single track crossing
- Sum signals to get
  - Total hit amplitude
  - Charge-weighted position centroid
- Currently use simple algorithm:
  - Look for local peak
  - When sequence "peak-valley-peak" is seen, split cluster at "valley"
  - Regardless of shape, limit clusters to a maximum size
- Improvements
  - Match hits by their pulse shape, *i.e.* timing centroid
  - Redo clustering after preliminary tracking (*e.g.* better cluster splitting)
  - . . . possibly more
- *NB:* Clustering does not necessarily have to be separate from tracking, could be integrated into a progressive tracking algorithm