SoLID Track Reconstruction

Ole Hansen

Jefferson Lab

SoLID Collaboration Meeting
July 10, 2014

SoLID Tracking Considerations

- High rates $\mathcal{O}(1\ \mathrm{MHz/cm^2})$, high occupancies (15%) $\rightarrow$ difficult environment
- Tracks not straight
- GEM coordinate axes not parallel (in current PVDIS design)
- Real-time tracking for level-3 trigger $\rightarrow$ want fast algorithm

Choice of Reconstruction Algorithm

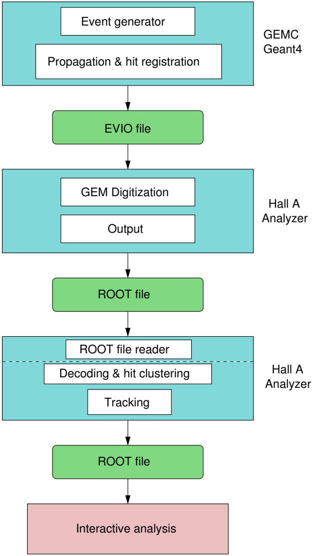
- Curved tracks, non-parallel coordinate axes → progressive algorithm (Kalman filter). Slow.
- Very preliminary version exists for SoLID (Xin Qian), being further developed by Duke group (Zhihong Ye, Weizhi Xiong).
- Little expertise in Hall A, but available in other halls → consult
- This is a multi-year development effort (but we have the time)

This Talk: Track Reconstruction Feasibility Study

- Simplify the problem in the simulation:
  - Rotate GEM strips in software $\rightarrow$ parallel coordinate axes
  - Simulate DIS signal without magnetic field $\rightarrow$ straight tracks
  - Background added separately $\rightarrow$ can vary background level
  - Expect this still to demonstrate *feasibility* of track finding
- Use existing TreeSearch reconstruction (BigBite)
  - Available now
  - Well tested & integrated in Hall A analyzer
  - Shown to work with SBS GEM trackers at $\geq$ SoLID occupancies

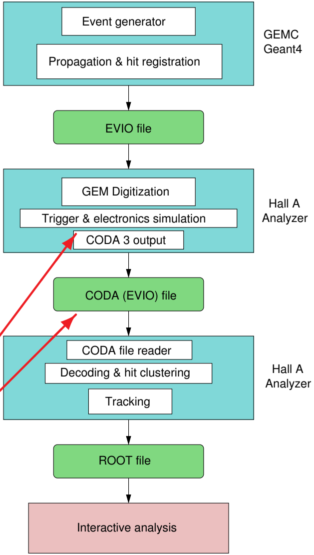
Track Reconstruction Simulation

- solgemc EVIO files as digitization input (S. Riordan)
- GEM digitization based on SBS work (E. Cisbani, R. Holmes)
  - APV25 pulse shape
  - Background added with randomized time offset
  - No other detectors digitized yet
  - Generated data (tracks, vertices) passed through
- ROOT file interface
- Tracking



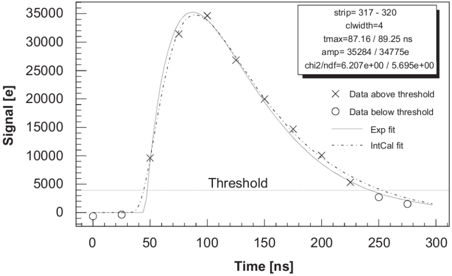
Track Reconstruction Simulation ("data challenge" ready)

- `solgemc` EVIO files as digitization input (S. Riordan)
- GEM digitization based on SBS work (E. Cisbani, R. Holmes)
  - APV25 pulse shape
  - Background added with randomized time offset
  - No other detectors digitized yet
  - Generated data (tracks, vertices) passed through
- ROOT file interface
- Tracking
- Should eventually use actual DAQ format (CODA 3) for analyzer input



APV25 Pulse Shape Deconvolution & Noise Filtering

S. Gadomski *et al.*, NIM A320, 217 (1992)



- For a first-order RC circuit, the original signal amplitudes s_k can be recovered from only three measured values v_k:

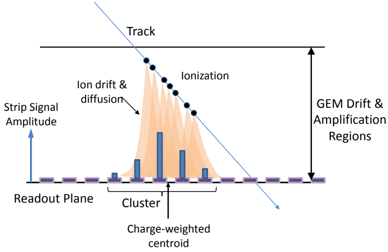
$$s_k = w_1 v_k + w_2 v_{k-1} + w_3 v_{k-2}$$

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

$$\text{Integrated amplitude:} \quad A \approx \sum_{k=1}^{3} s_k$$

- 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

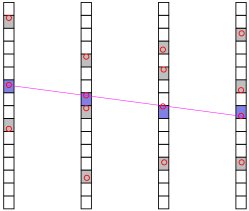


- Signals on adjacent readout strips typically belong to a single track crossing
- Sum signals to get
  - total hit amplitude
  - charge-weighted position centroid
- Currently use simple algorithm:
  - Look for local peak
  - When sequence "peak-valley-peak" is seen, split cluster at "valley"
  - Regardless 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

TreeSearch Algorithm

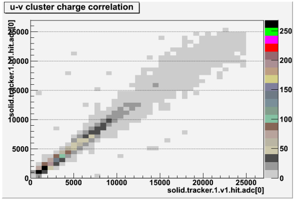
M. Dell'orso and L. Ristori, NIM A287, 436 (1990)

- Recursive template matching algorithm (global, non-progressive)
- Advantages
  - Very fast ($\mathcal{O}(\log N)$) and memory-efficient ($\mathcal{O}(10\text{ MB})$)
  - Independent of other detectors $\rightarrow$ no seed needed

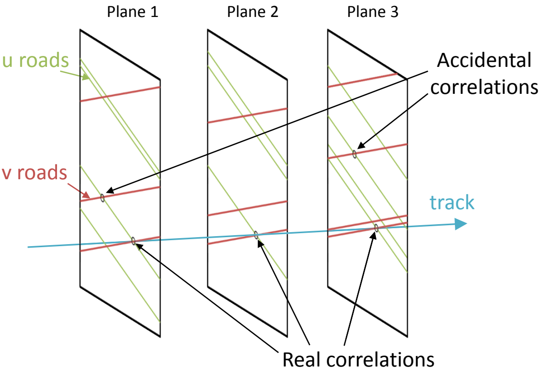


- Limitations
  - Works in 2D only (one readout coordinate, "projection")
  - Only suitable for straight tracks
- Used by HERMES, Qweak, etc.

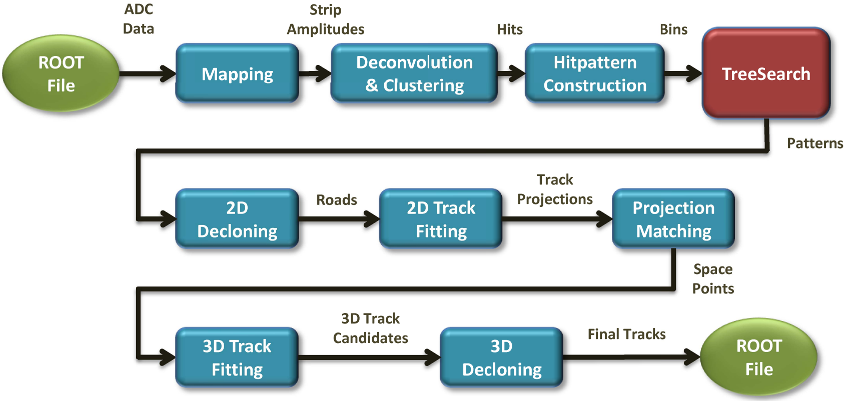
3D Matching



- Correlate roads from different projections via hit amplitude in shared readout planes
- Pair roads with the best overall correlation to get space points for 3D track fits
- Calorimeter hit(s) can help resolve ambiguities



TreeSearch Track Reconstruction Chain (GEM version)



FYI: Progress Since Last Meeting

- Digitized 5-GEM data set ("muon, no field, all materials" from August 2013) with up to 100% background level (see later)
- Fixed major bug causing previously seen low tracking efficiency (see later)
- Implemented generic "simulation decoder" interface in Hall A analyzer, providing access to MC truth information
  - Reconstructed hits now directly include MC information (position, time, signal/background status etc.) $\rightarrow$ accuracy check
  - Prerequisite for flagging ghost tracks
  - GEM hits of primary MC track now associated with reconstructed hits $\rightarrow$ efficiency check
  - Plane-to-plane evolution of MC track parameters now easily accessible $\rightarrow$ energy loss, ToF, angular deflection
- Uploaded all relevant source code to Github:
  - `JeffersonLab/TreeSearch` repository, `solid` branch
  - `JeffersonLab/libsolgem` repository, `ole` branch
  - Replay scripts and databases to be checked in shortly

MC Data Sets

(simulations by Seamus Riordan)

- Configuration
  - 40 cm LD_2 target in 11 GeV beam
  - PVDIS detector setup with 5 GEM planes
  - baffles (which?)
- "Signal Runs"
  - Generator: DIS
  - Only interactions of primary particle recorded
  - Available data sets

Primary particle	Field	Materials	# events w/GEM hits
μ^-	off	trackers	240k
μ^-	off	"all"	248k
e^-	on	"all"	14k

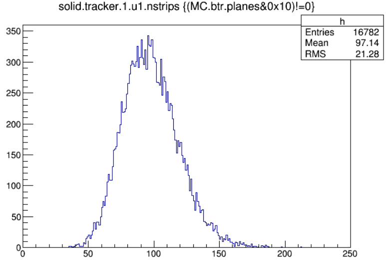
  - "Trackers" materials: only interactions with GEM trackers recorded ("ultra-clean data")

Background Simulation

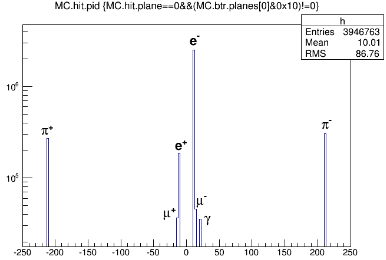
- "Background Runs"
  - Same configuration as for signal runs, except field always on
  - Simulated 198 M background events (= electrons passing through target)
  - Production rate $\approx$ 40 M/hr
- Adding background to signal runs in digitization step
  - $\approx$ 86 M electrons pass through target in a 275 ns time window at 50 μA $\equiv$ 100% background
  - To reduce analysis time: fold background from 30 sectors into signal sector with random time offset per sector
  - Obviously not enough background events for any significant number of signal events $\rightarrow$ re-use events, but with different time randomization
- Status
  - 100k signal events from "μ^-/field off/all materials" digitized with 0%, 1%, 5%, 10%, 25%, 50% and 100% background
  - At 100% bg level, digitization rate is $\approx$ 500 signal events/hr/CPU core, or $\approx$ 7 CPU core seconds/event (on Xeon E5-2650v2 "Ivy Bridge")

Strip Occupancy, 100% Background, 5 GEM Planes

Number of strips above ADC threshold *after* noise cut



PID of hits



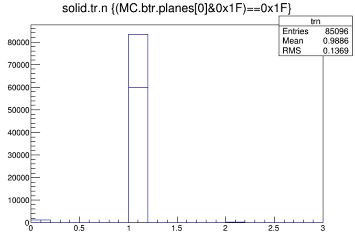
Filtered occupancies, 5 GEM setup, 100% background			
Plane	Mean # active	Total # strips	Occupancy (%)
u1	97.1	753	12.9
v1	105.4	627	16.8
u2	93.4	945	9.9
v2	95.1	659	14.4
u3	87.4	921	9.5
v3	91.7	657	14.0
u4	76.7	1271	6.0
v4	79.3	1271	6.2
u5	75.4	1309	5.8
v5	78.6	1309	6.0

- First plane sees many slow electrons ($p < 1$ MeV)
- *u*-*v* asymmetry not fully understood yet
- Estimated SBS raw occupancy $< 20\%$ in all planes
  $\rightarrow$ estimated SoLID occupancies below SBS

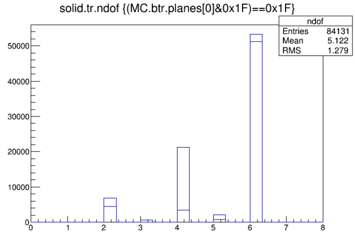
Tracking Efficiency For No Background

"Muons, no field, all materials" data set

Number of tracks found for MC tracks crossing all planes



Degrees of freedom of track fits (max = 6)



- Track finding efficiency ("any track")

$$\frac{83807}{85096} = 98.5\%$$

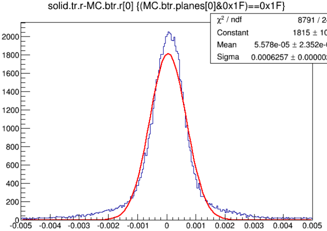
- If requiring "accurate" reconstruction (all residuals within 3σ of MC track):

$$\frac{60059}{85096} = 71\%$$

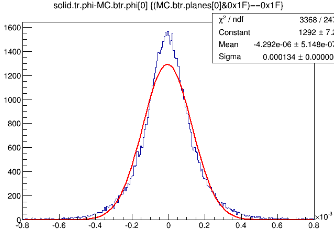
- Neither definition satisfactory
  - With real data, even "not accurate" tracks may appear good
  - All of the found tracks in an event may be garbage
  - Need to apply realistic selection criteria, *e.g.* vertex cut, match to calorimeter hit
- Ghost and background track rates not yet determined. (Analysis code almost complete.)

True Residuals (No Background, 5 GEM Planes)

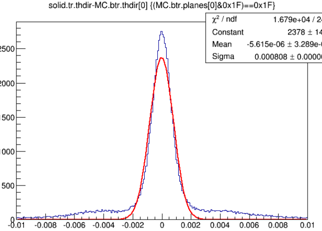
r-coordinate of crossing point in first GEM plane



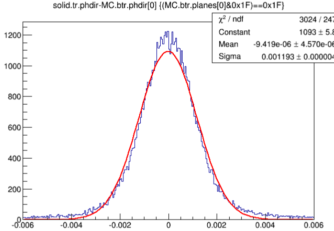
ϕ-coordinate of crossing point in first GEM plane



θ_{dir}: Polar angle of momentum

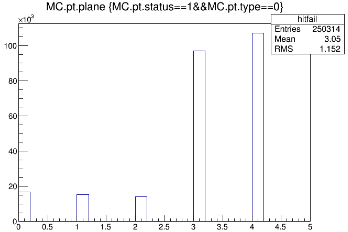


ϕ_{dir}: Azimuth of momentum

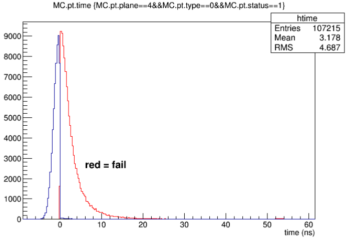


FYI: Fixing THE Bug (bad test in APV25 noise filter)

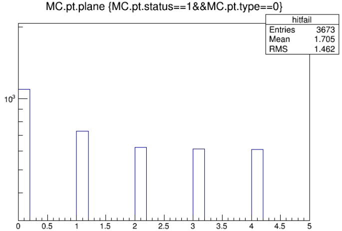
Plane of primary μ hits without a reconstructed hit



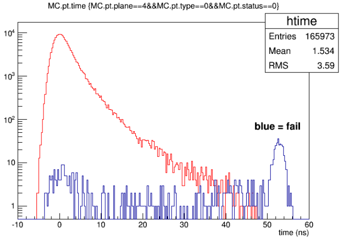
Relative time of primary μ hit at plane 4



As above, after bugfix

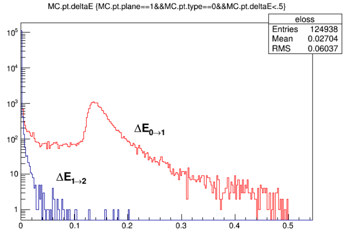


As above, after bugfix

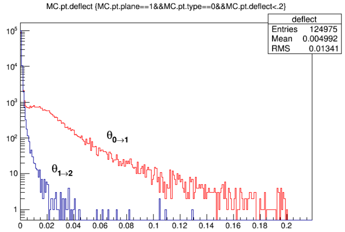


FYI: MC Data Oddity

Energy loss of primary μ between trackers (GeV)



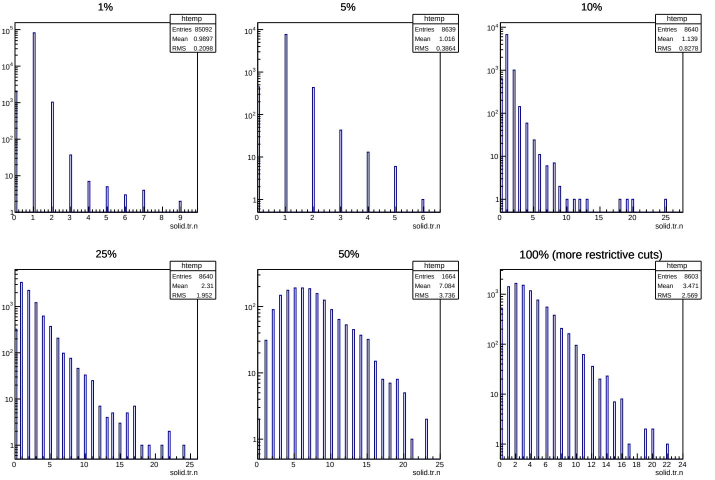
Deflection angle of primary μ between trackers (rad)



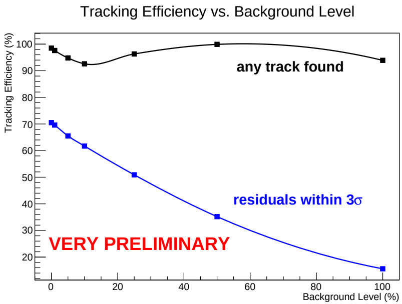
Looks like some material protrudes into the GEM acceptance between the first and second tracker plane

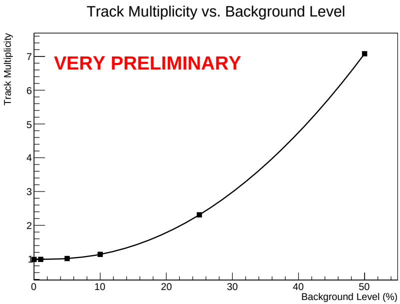
Tracking Performance With Background

"Muons, no field, all materials" data set



Tracking Performance vs. Background Level





Conclusions

- Tracking efficiency at $\leq 10\%$ background levels reasonable: $> 90\%$.
- Residuals have noticable tails even with clean data. Reconstruction algorithm and parameters still need fine-tuning. Results are preliminary. Nevertheless, it looks like the showstopper bugs have been found.
- Higher background levels challenging. High track multiplicities, low rate of accurately found tracks. Currently unclear if tracking feasible.
- At least roughly realistic data analysis setup would be useful: calorimeter cut, vertex reconstruction, etc.
- Ghosts and background track rates to be evaluated
- NB: Background may be underestimated due to presence of field. TreeSearch rejects curved tracks.
- FYI: Implications of found bugs and surprising results of MC truth data comparisons should be evaluated by SBS collaboration as well

Outlook

- Next 12–18 months:
  - Re-run simulation with latest design, if needed (?)
  - Debug/finish feasibility study with TreeSearch.
  - Include other relevant detectors in digitization & analysis
  - Tentative: Make GEM digitization more realistic (cluster size too small)
  - Demonstrate tracking at 100% background level, better yet 200%
  - Additional *competent* manpower highly welcome.
- Longer term:
  - Develop progressive tracking algorithm
  - Learn from Halls B & D
  - Demonstrate curved track reconstruction feasibility, performance etc.

General Thoughts on SoLID Analysis Software

- Main item: Choice of software framework
- SoLID could be *at least* as data-intensive as Hall D!
- Some choices
  - Hall A analyzer. Collaborators are familiar with it. Not parallelized, but that may not be a show-stopper (and can be added). But: not tested with very large data sets. No support yet for pipelined electronics. Low manpower.
  - Hall D framework (JANA). Fully parallel, tested with huge data sets. C++-based.
  - Hall B (CLARA). Not sure how easily adaptable to non-CLAS apparatuses. Mostly Java-based, with support for C++ modules. Massive scalability, tested with huge data sets. Software-as-a-service model unfamiliar in Hall A.
- Should decide at least 4–5 years before start of running
- Should begin thinking of forming a analysis software group