Hall A Analysis Software Status Report

Ole Hansen

Jefferson Lab

Joint Hall A & C Data Analysis Workshop
December 12, 2012

Hall A Analysis Software: Podd (C++ Analyzer)

- Version 1.5 in production since 2008
  - Used by every experiment (except for parity in integrating mode)
  - Very stable
- BUT need to think of 12 GeV future
  - Recent software review
  - Some improvements desirable to support 12 GeV experiments
  - Collaboration with Hall C started to develop common software based on Podd
  - Plenty of work ahead

Home page: `http://hallaweb.jlab.org/podd/`

Podd 1.5: 2011–12 Improvements

- Maintenance
  - Lots of small bugfixes & 64-bit compatibility fixes
  - Support for latest ROOT, Linux and compiler versions
- New features
  - Support for Qweak helicity scheme analysis (J. Roche)
  - Support for additional VME frontend modules (L. Selvy)
  - Preliminary Mac OS X support
- Code development
  - Moved to `git` version control system

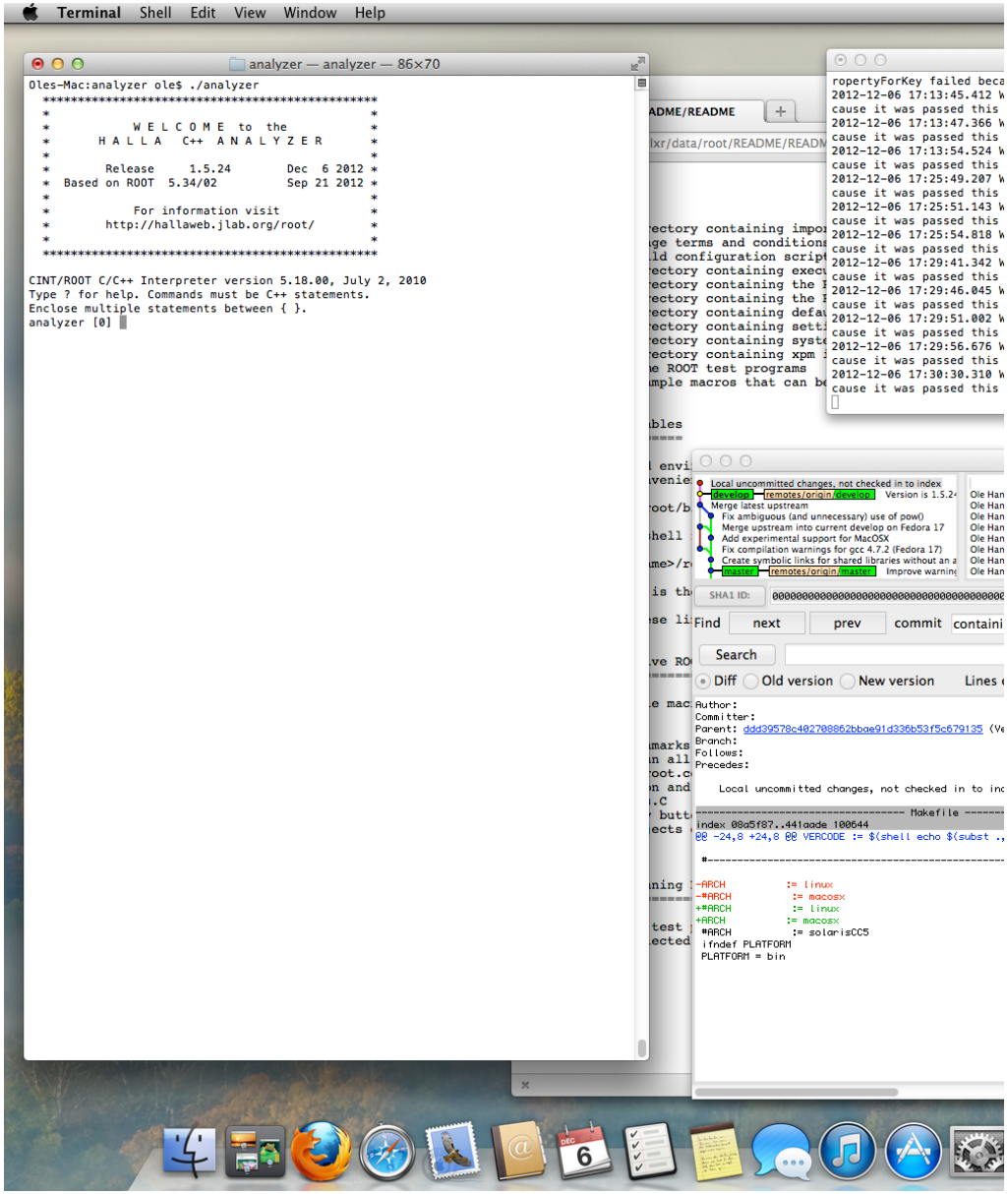
Current version: 1.5.24 web

Mac OS X Support (experimental)

Getting started:

- Install `Xcode` with command line developer tools
- Install pre-build OSX version of ROOT from CERN & run included `thisroot.sh`
- On OSX $\geq$ 10.7, install `XQuartz` for X11 support
- Set `ARCH=macosx` in top-level `Makefile`
- Build as usual (`make -j3`)
- Set `DYLD_LIBRARY_PATH`

Feedback welcome!



Version Control with git

- Podd under CVS since 2001
- Today, CVS is obsolete (many limitations, awfully slow) → replace
- git suggested by Hall C. Turns out to be quite impressive
- Podd repository converted to git in October. Full CVS history kept.
- Onsite access only for now. Will set up a server shortly.

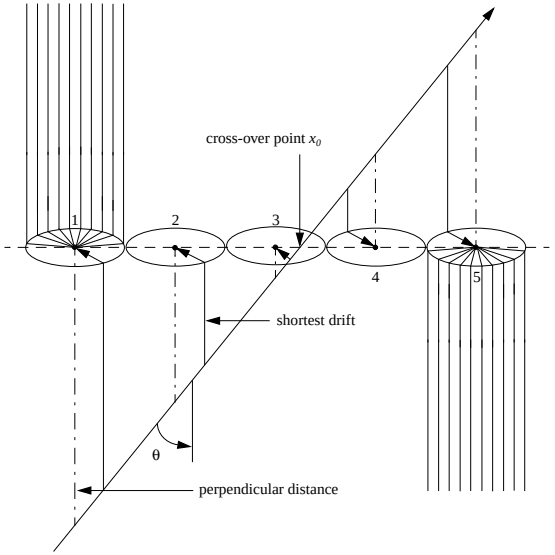
Cloning the Podd Repository (onsite)

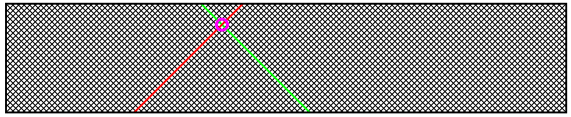
```
[ole@fedora17 Develop]$ git clone jlabl3:/group/halla/analysis/git/analyzer.git
Cloning into 'analyzer'...
remote: Counting objects: 8813, done.
remote: Compressing objects: 100% (2360/2360), done.
remote: Total 8813 (delta 6462), reused 8782 (delta 6441)
Receiving objects: 100% (8813/8813), 3.15 MiB | 4.21 MiB/s, done.
Resolving deltas: 100% (6462/6462), done.
```

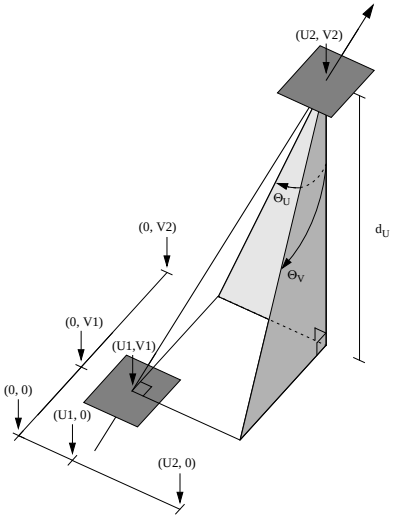
See also Mark Jones's tutorial this afternoon ▸ talk

VDC Tracking Revisited: Algorithm

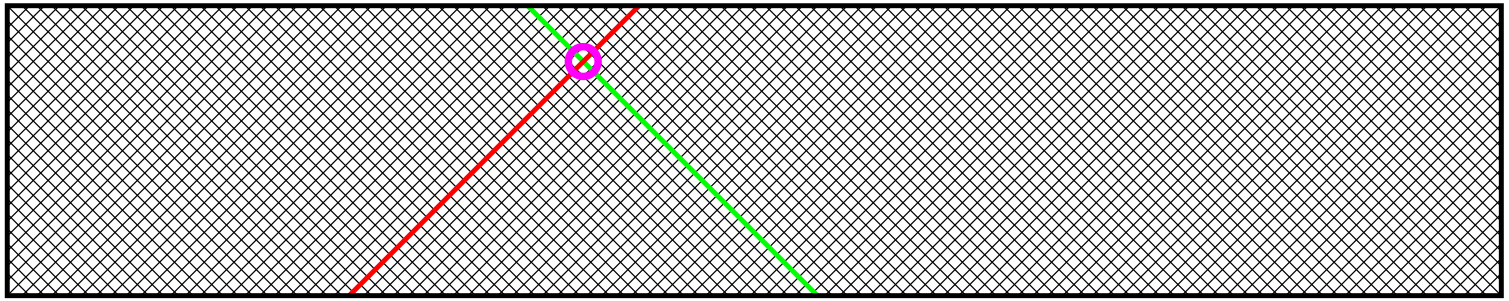
- Find clusters in each plane, fit to drift distances & obtain crossover points
- Match u and v clusters in each chamber
- Connect matched points in top and bottom, requiring consistent track angles
- Reconstruct target quantities using reverse transport matrix







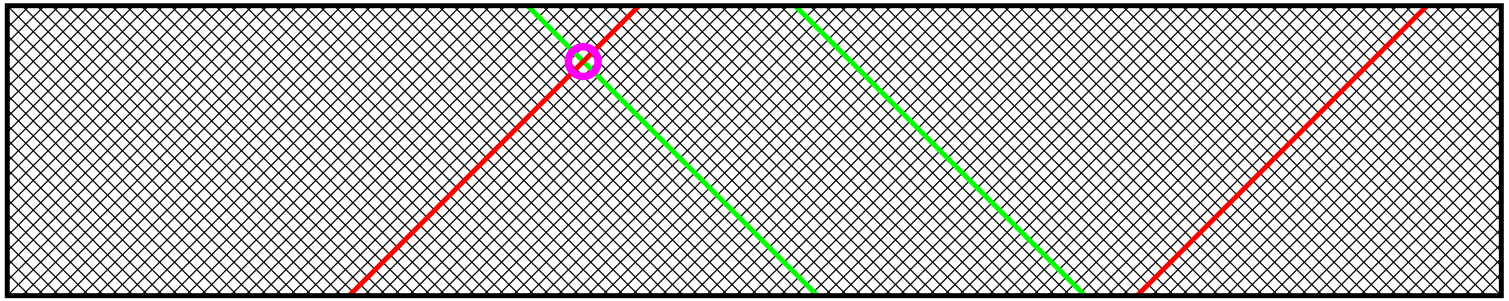
VDC Tracking: u and v Cluster Matching



Goal: Match u and v clusters in each chamber

- Obvious if only one cluster per plane
- Multiple clusters are *often* not a problem
- Ambiguities arise if several clusters are close to each other
- Poor man's solution: just throw out ambiguous events (ok at low rate)
- Ways to disambiguate:
  - Use 3rd wire direction
  - Correlate track candidates with signals in other detectors
  - Take advantage of cluster timing

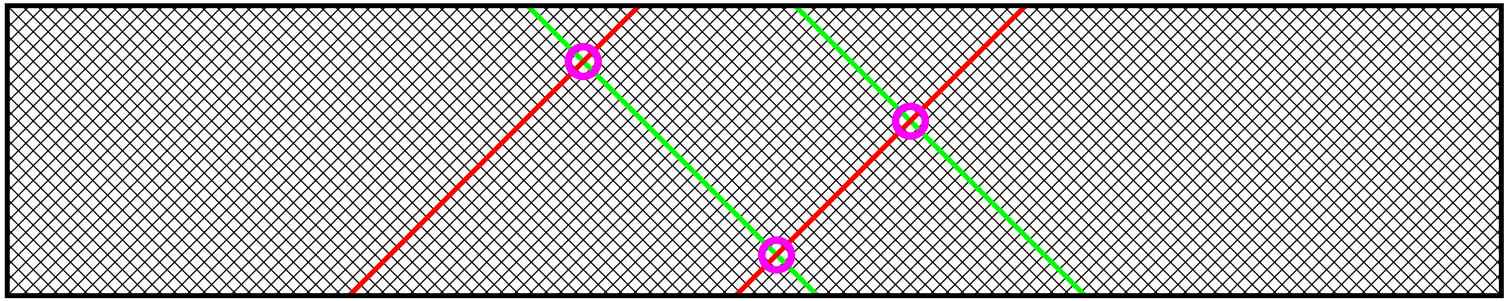
VDC Tracking: u and v Cluster Matching



Goal: Match u and v clusters in each chamber

- Obvious if only one cluster per plane
- Multiple clusters are *often* not a problem
- Ambiguities arise if several clusters are close to each other
- Poor man's solution: just throw out ambiguous events (ok at low rate)
- Ways to disambiguate:
  - Use 3rd wire direction
  - Correlate track candidates with signals in other detectors
  - Take advantage of cluster timing

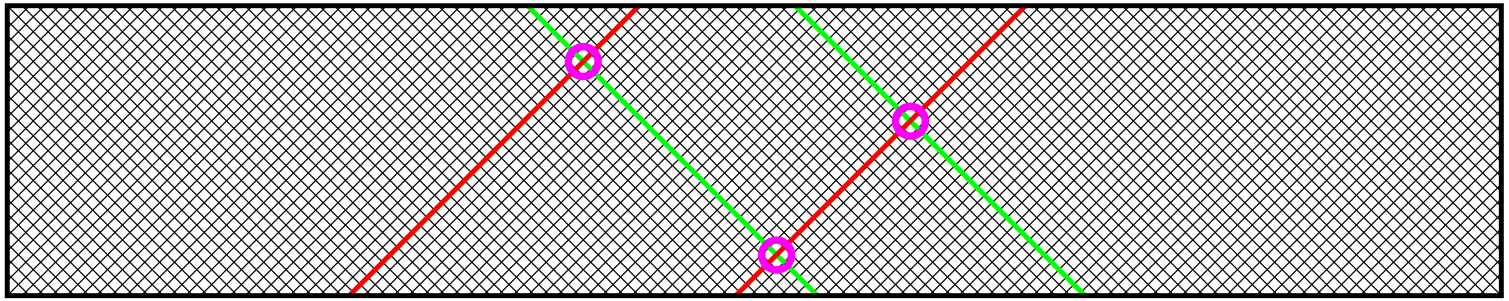
VDC Tracking: u and v Cluster Matching



Goal: Match u and v clusters in each chamber

- Obvious if only one cluster per plane
- Multiple clusters are *often* not a problem
- Ambiguities arise if several clusters are close to each other
- Poor man's solution: just throw out ambiguous events (ok at low rate)
- Ways to disambiguate:
  - Use 3rd wire direction
  - Correlate track candidates with signals in other detectors
  - Take advantage of cluster timing

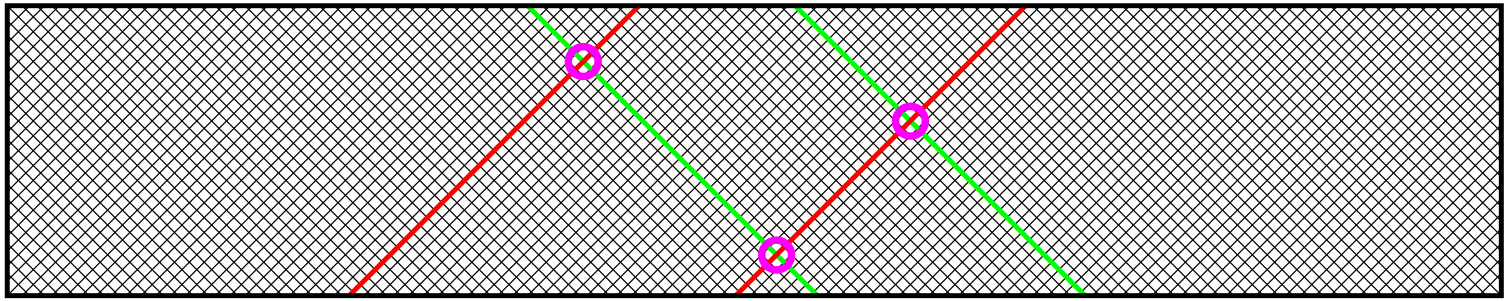
VDC Tracking: u and v Cluster Matching



Goal: Match u and v clusters in each chamber

- Obvious if only one cluster per plane
- Multiple clusters are *often* not a problem
- Ambiguities arise if several clusters are close to each other
- Poor man's solution: just throw out ambiguous events (ok at low rate)
- Ways to disambiguate:
  - Use 3rd wire direction
  - Correlate track candidates with signals in other detectors
  - Take advantage of cluster timing

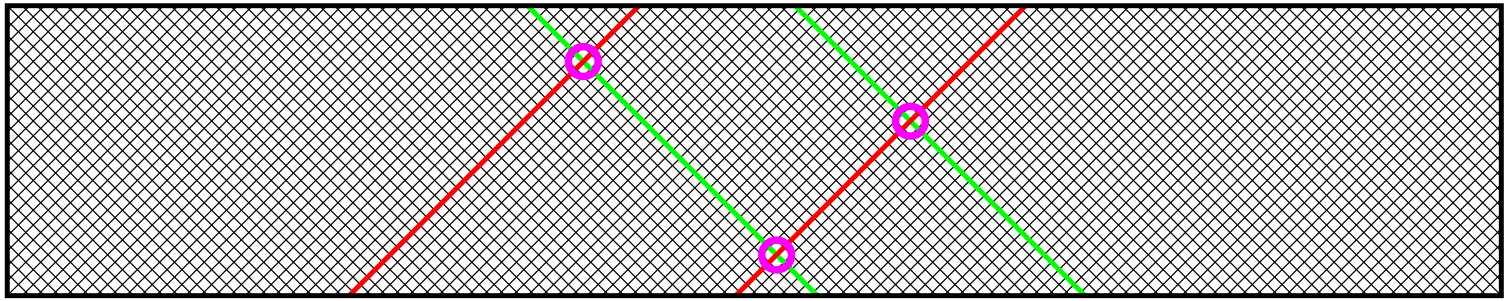
VDC Tracking: u and v Cluster Matching



Goal: Match u and v clusters in each chamber

- Obvious if only one cluster per plane
- Multiple clusters are *often* not a problem
- Ambiguities arise if several clusters are close to each other
- Poor man's solution: just throw out ambiguous events (ok at low rate)
- Ways to disambiguate:
  - Use 3rd wire direction
  - Correlate track candidates with signals in other detectors
  - Take advantage of cluster timing

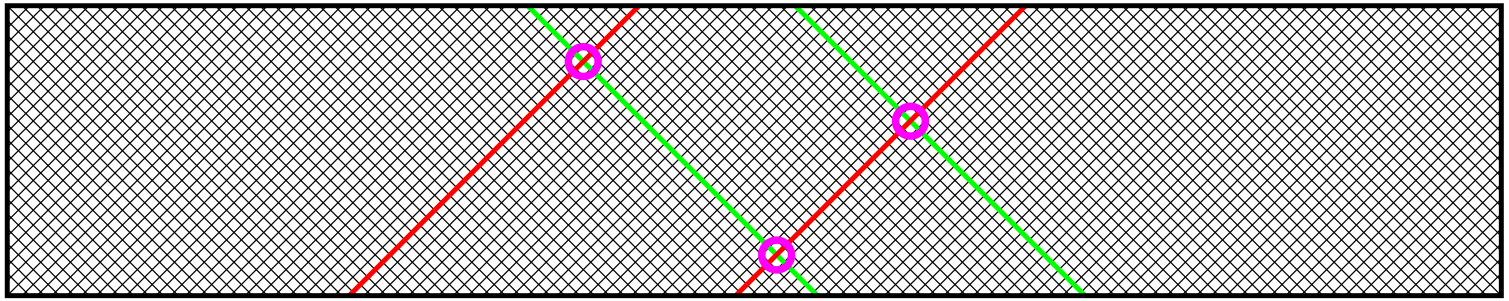
VDC Tracking: u and v Cluster Matching



Goal: Match u and v clusters in each chamber

- Obvious if only one cluster per plane
- Multiple clusters are *often* not a problem
- Ambiguities arise if several clusters are close to each other
- Poor man's solution: just throw out ambiguous events (ok at low rate)
- Ways to disambiguate:
  - ~~Use 3rd wire direction~~ $\rightarrow$ expensive upgrade
  - Correlate track candidates with signals in other detectors
  - Take advantage of cluster timing

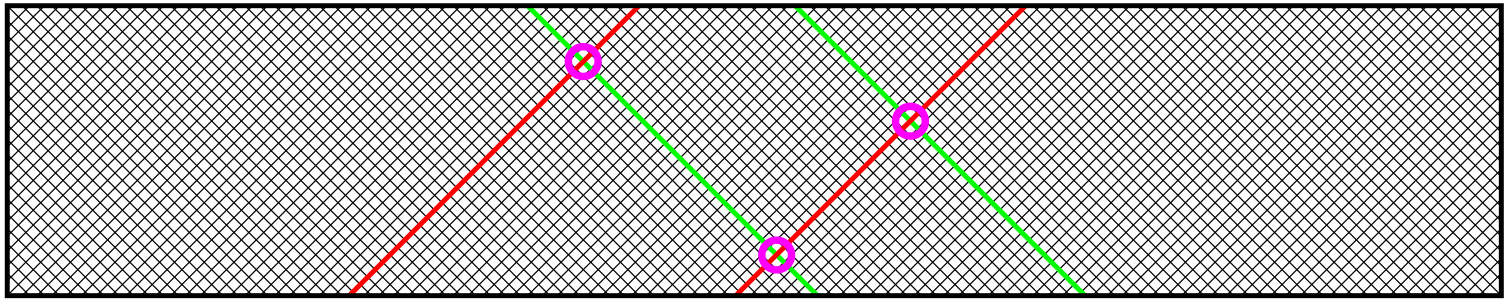
VDC Tracking: u and v Cluster Matching



Goal: Match u and v clusters in each chamber

- Obvious if only one cluster per plane
- Multiple clusters are *often* not a problem
- Ambiguities arise if several clusters are close to each other
- Poor man's solution: just throw out ambiguous events (ok at low rate)
- Ways to disambiguate:
  - ~~Use 3rd wire direction~~ $\rightarrow$ expensive upgrade
  - Correlate track candidates with signals in other detectors
  - Take advantage of cluster timing

VDC Tracking: u and v Cluster Matching

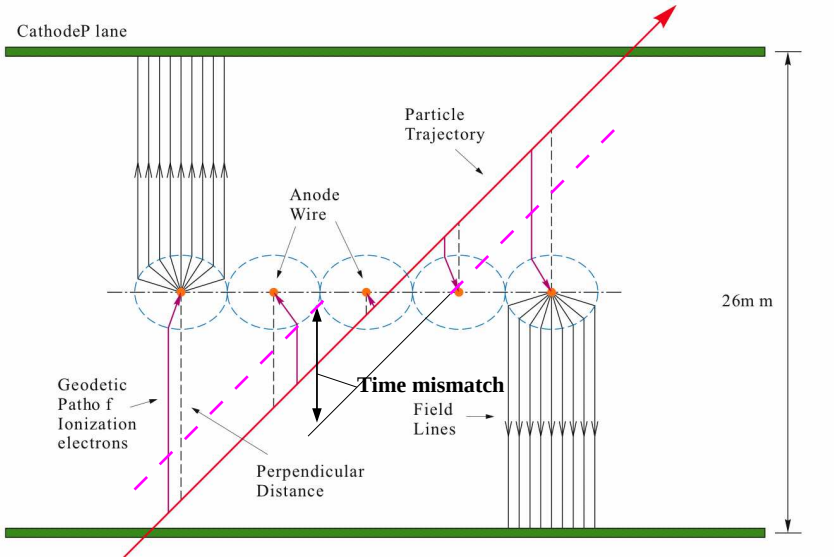


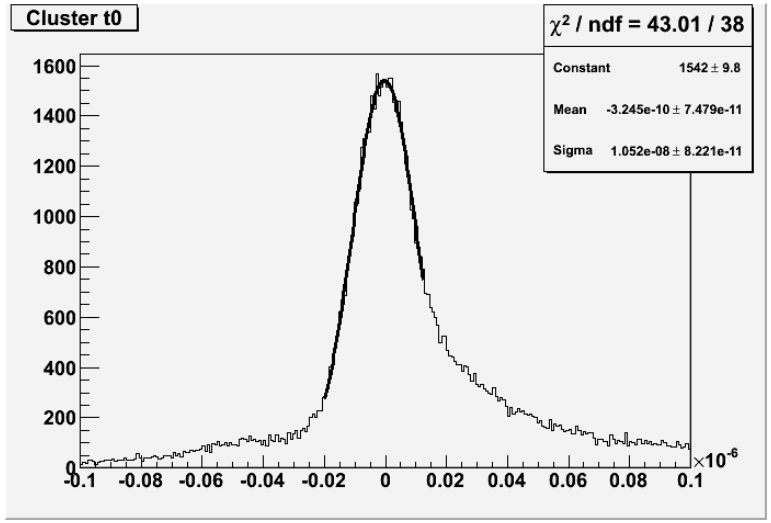
Goal: Match u and v clusters in each chamber

- Obvious if only one cluster per plane
- Multiple clusters are *often* not a problem
- Ambiguities arise if several clusters are close to each other
- Poor man's solution: just throw out ambiguous events (ok at low rate)
- Ways to disambiguate:
  - ~~Use 3rd wire direction~~ $\rightarrow$ expensive upgrade
  - Correlate track candidates with signals in other detectors
  - Take advantage of cluster timing

Multi-cluster VDC Track Reconstruction

Developed for 2010 APEX test run: $\mathcal{O}(1\ \text{MHz})$ rate, ≈ 2 accidental tracks per event





- Non-linear 3-parameter fit to extract track time offset t_0 (M. Paolone)
- 20 ns FWHM time resolution $\rightarrow$ background rejection factor $\approx$ 5–10
- Match u and v clusters with $t_0 \approx 0$ (trigger track) $\rightarrow$ Resolves ambiguous events in at least $\approx 70\%$ of the cases
- Should try to resolve remaining ambiguities with other info, *e.g.* scintillator hits
- Downside: about $\times 20$ slower than 2-parameter fit. Alternative fast fitting code exists (summer student project), similar results, but at prototype stage
- Not yet in Podd distribution. Needs $\approx$ 1 month's worth of finishing work.

June 2012 Software Review

- External review of 12 GeV software preparations of all halls
- Generally very positive
- Suggestions for Hall A
  - Make use of good development practices as much as possible
    - Nightly builds
    - Standard code evaluation tools (memory leak checkers etc.)
    - Validation procedures (reference results, test suites etc.)
  - Parallelize Podd
  - Find limits of SBS track reconstruction & improve if necessary
  - Collaborate with Hall C
- Annual follow-up reviews planned. Next: ca. September 2013
- Expect increased attention to software as we approach 12 GeV

Collaboration with Hall C

- Already in late 2011, Hall C decided to adopt Hall A C++ Analyzer framework as basis for a rewrite of the Hall C analysis software for 12 GeV → see Gabi's talk, up next
- Hall A/C collaboration also recommended by review committee to improve sharing of software between halls
- Bi-weekly meetings
- Has already implicitly sparked a thorough code review of both the Hall A and C analyzers
- Changes necessary in Podd
  - Split libraries into "core" and Hall A-specific parts
  - Add in Hall C-specific changes to detector classes ("hit lists" etc.)
  - Write interface for Hall C-style database
  - Also a good time for other, relatively easy improvements → next slide
- The resulting updated analyzer will become Release 1.6

Roadmap I: Podd 1.6 ("Collaborative edition")

Planned Features

- Code split into core and hall-specific libraries
- Hall C changes integrated
- Object-oriented decoder
- "Event Type Handler" plug-ins
- Abstract database interface for all modules, similar to existing `LoadDB`
- Time-zone safe `TTimeStamp` for all dates/times
- Backlog of binary-compatibility workarounds removed
- Test & validation procedures, as recommended in the review
- Bug-tracking system (maybe)

A few things started, none finished. Expect several months of work.

Roadmap II: Podd 2.0 ("12 GeV edition")

Wishlist

- Automatic event-level parallelization → major upgrade
- Significant ROOT file output speed improvement, if at all possible
- Decoders for pipelined 12 GeV electronics. CODA 3 support.
- SQL database backend. (Expect databases to become big and complex.)
- VDC multicluster analysis

Lots of work. We could absolutely use help

Rel	Task	Status	Time (weeks)	Sum
1.6	Split libraries into core and Hall A parts	Not started	2	
	Integrate Hall C changes	Not started	1	
	Object-oriented decoder	Not started	4	
	Event-type handler plugins	About 50% done	4	
	Abstract database API design	About 80% done	1	
	Modify all modules for new database API	Not started	2	
	Change all classes to TTimeStamp	Not started	1	
	Remove binary-compatibility workarounds	Not started	1	
	Implement test & validation procedures	Partly done (25%?)	4	
	Set up bug-tracking system	Not started	2	22
2.0	Automatic parallelization	Researched (10%)	10	
	ROOT output file speed improvements	Not started	6	
	Decoders for 12 GeV electronics	Some (10%?)	8	
	CODA 3/EVIO 3 support	Not started	2	
	SQL database backend	About 25% done	4	
	VDC multicluster analysis	About 75% done	3	33

Software for Specific Experiments

- G_M^p wants to use an FPP chamber in the HRS to assist tracking
- DVCS: hopefully have all their software (?)
- APEX: if scheduled, would definitely need VDC multicluster analysis
- SBS (SuperBigBite)
  - GEM track reconstruction optimization
  - Coordinate detector analysis
  - GEp(5) recoil polarimetry
  - Calorimeter cluster search
  - RICH analysis & PID
- Møller is getting into early software engineering stage. Might need to optimize existing parity code for large data volume.
- SoLID
  - Heavy simulation activities, will be ongoing for years
  - Need to demonstrate track reconstruction feasibility

List is most certainly incomplete

Conclusions

- The theme for the next 1–2 years will be planning and upgrading our core analysis software for the requirements of 12 GeV running
- Long list of tasks, both backlog of old projects and upgrades driven by 12 GeV
- Additional manpower very desirable to keep schedule!
- Collaboration with Hall C is very promising. Hopefully will speed up development and increase feedback