BigBite Track Reconstruction

Ole Hansen

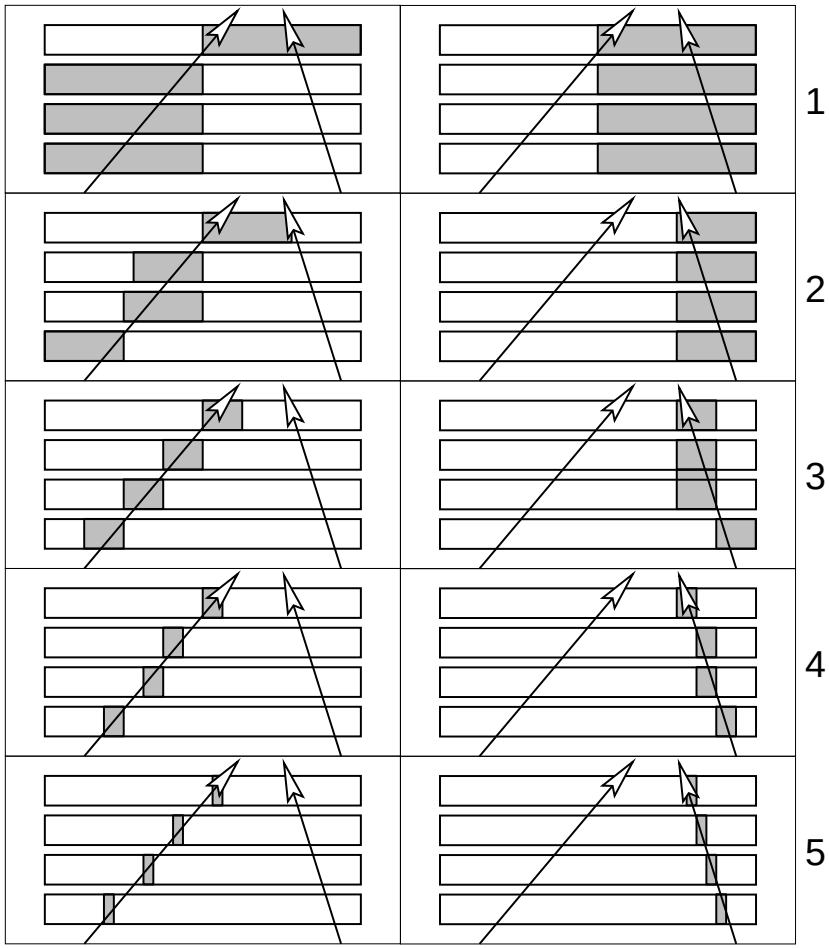
Jefferson Lab

Hall A Data Analysis Workshop
June 12, 2008

Tree Search Algorithm

- Suggested by Dell'orso *et al.*, NIM, 1990
- Recursive template matching
- Fast and efficient (speed and memory)
- Suitable for BigBite
  - simple geometry
  - field-free tracking region
- Proven at HERMES with chambers similar to BigBite's

Successive Approximation Method



Key Ideas

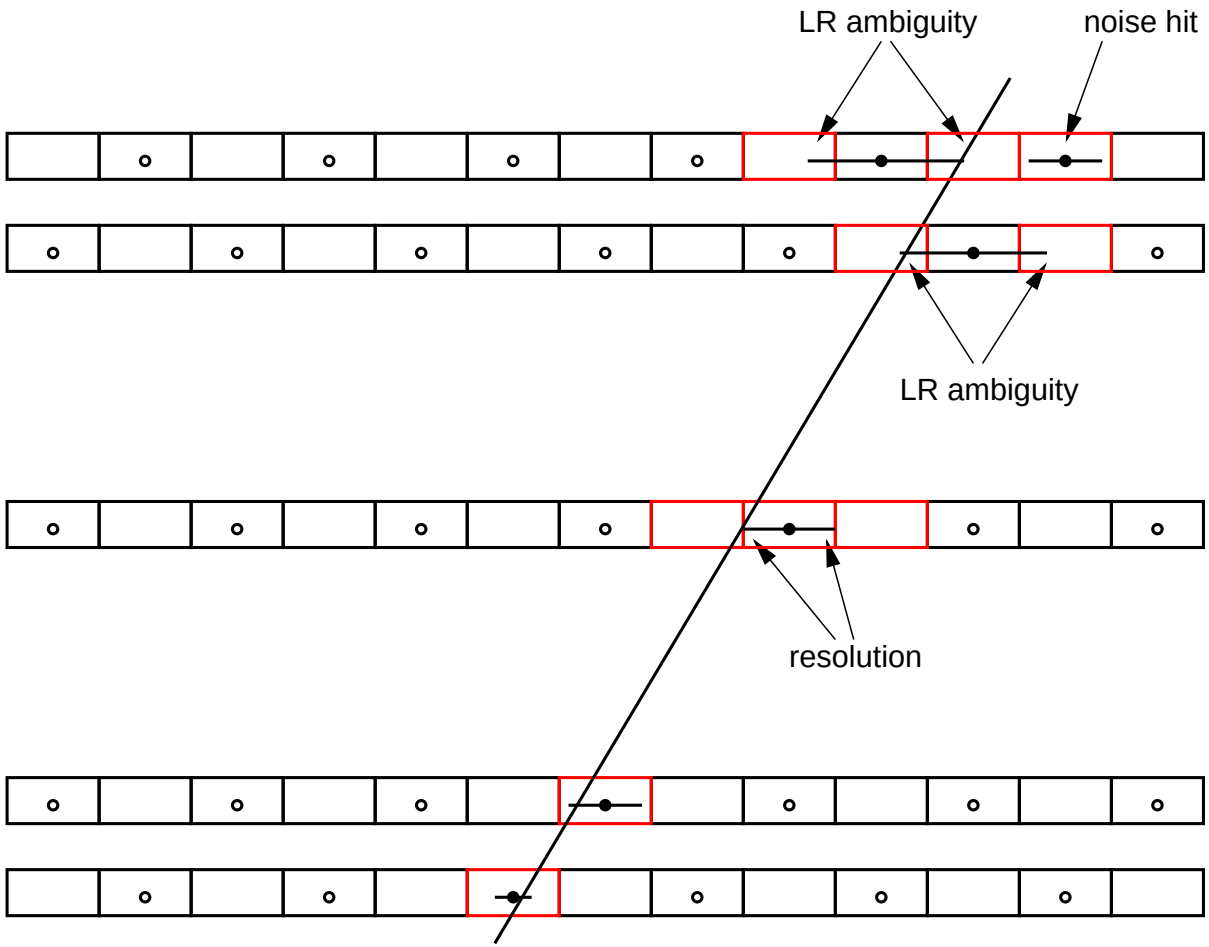
- Tree-like structure allows fast template lookup ($\mathcal{O}(\log N_{\text{bins}})$)
- Symmetry considerations allow efficient template storage ($\mathcal{O}(1\text{MB})$): only “base patterns” stored

Algorithm already discussed at last meeting

Finished Since Last Meeting

- 2D pattern "de-cloning" algorithm ($\rightarrow$ "roads")
- Track fitting within roads
- Combination of roads in 3D
- 3D track fitting
- 3D track de-cloning/de-ghosting algorithm
- Testing & debugging with online E04-007 data
- Adapted to new event display
- 2D processing parallelized
- Profiling & speed optimization ($\times 2$)

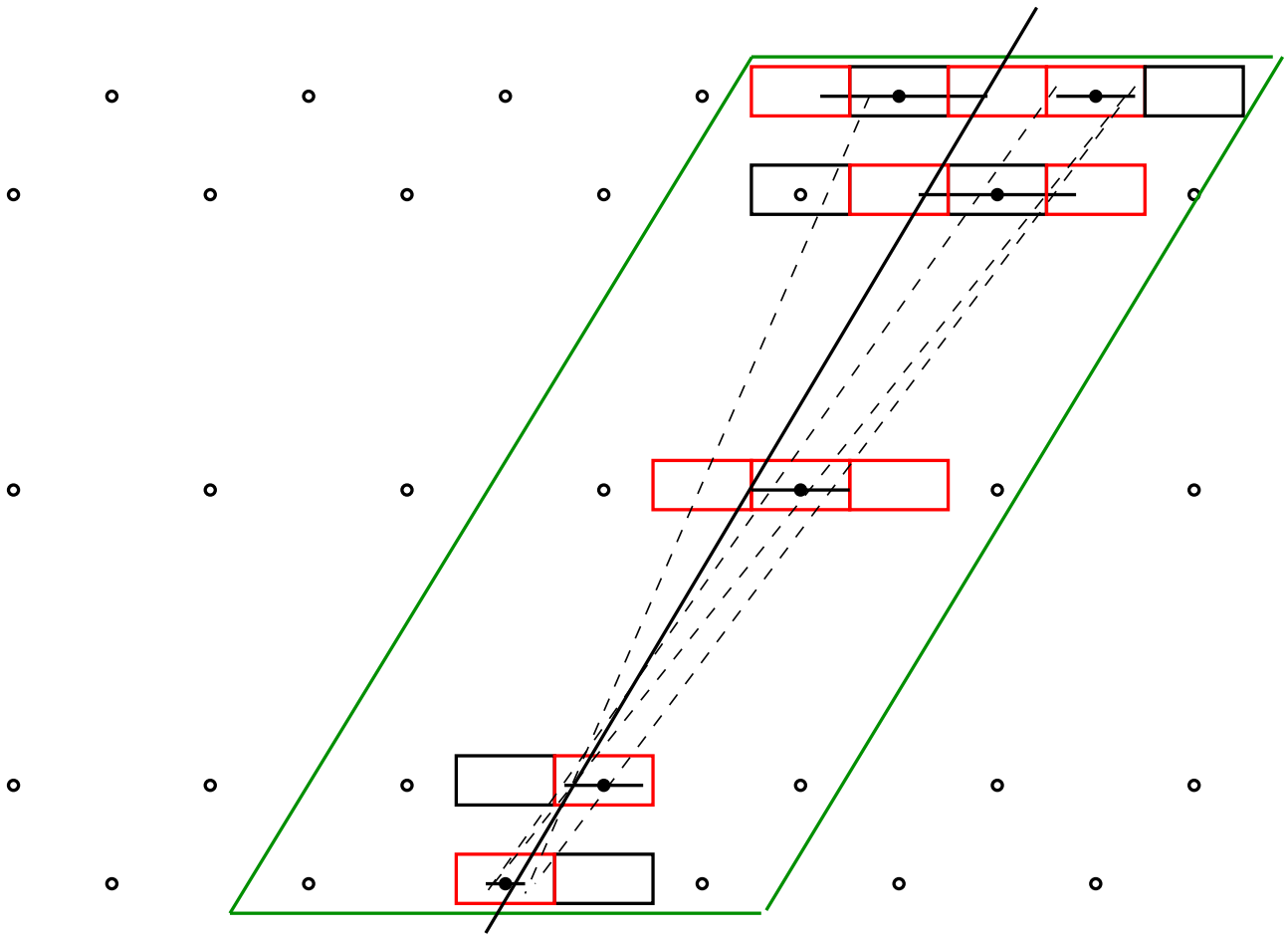
Pattern Cloning (multiple patterns for single track)



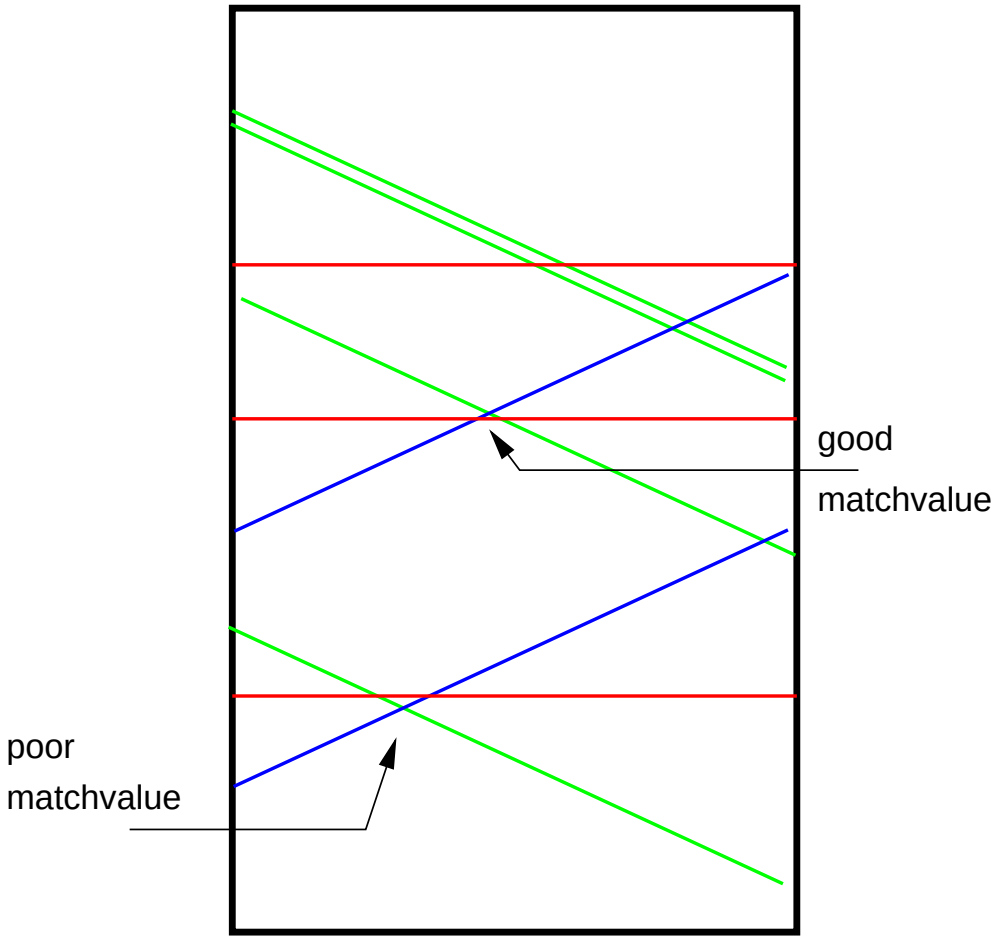
2D De-Cloning Algorithm

- "Clustering" problem
- Start with patterns with hits in the most number of planes ("cluster centers")
- Add patterns around start pattern whose set of hits is a subset of the start hit set (but allow extra hits if shared hit present in a given plane) ("similarity measure")
- Configurable: also allow patterns with neighboring hits
- Road boundaries defined by outermost bins set
- Time-critical
- Runs in $\approx N \log N$ time. Can probably be improved.

2D Track Fitting



Combination in 3D



3D Track Fitting

Fit the linear equations

$$\mathbf{A}\hat{\beta} = \mathbf{y}$$

using the coordinates y_i of the best 2D fits in all planes i,

$$y_i = \begin{pmatrix} x + m_x z_i \\ y + m_y z_i \end{pmatrix} \cdot \begin{pmatrix} \cos\alpha_i \\ \sin\alpha_i \end{pmatrix},$$

where x, m_x, y, m_y are the parameters to be fitted (β_k).
The fit is done by Cholesky decomposition of the normal equation

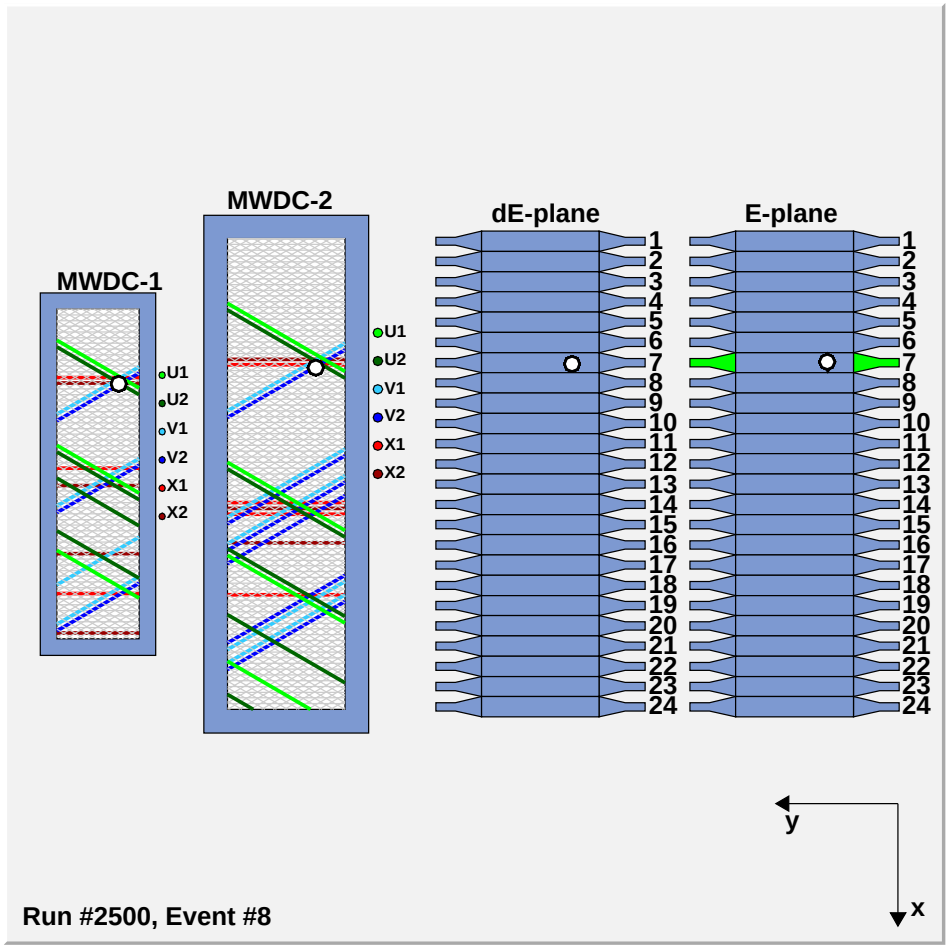
$$(\mathbf{A^T W A})\hat{\beta} = (\mathbf{A^T W})\mathbf{y}$$

where $\mathbf{W}$ is the weight matrix (cf. ROOT's `TLinearFitter`).

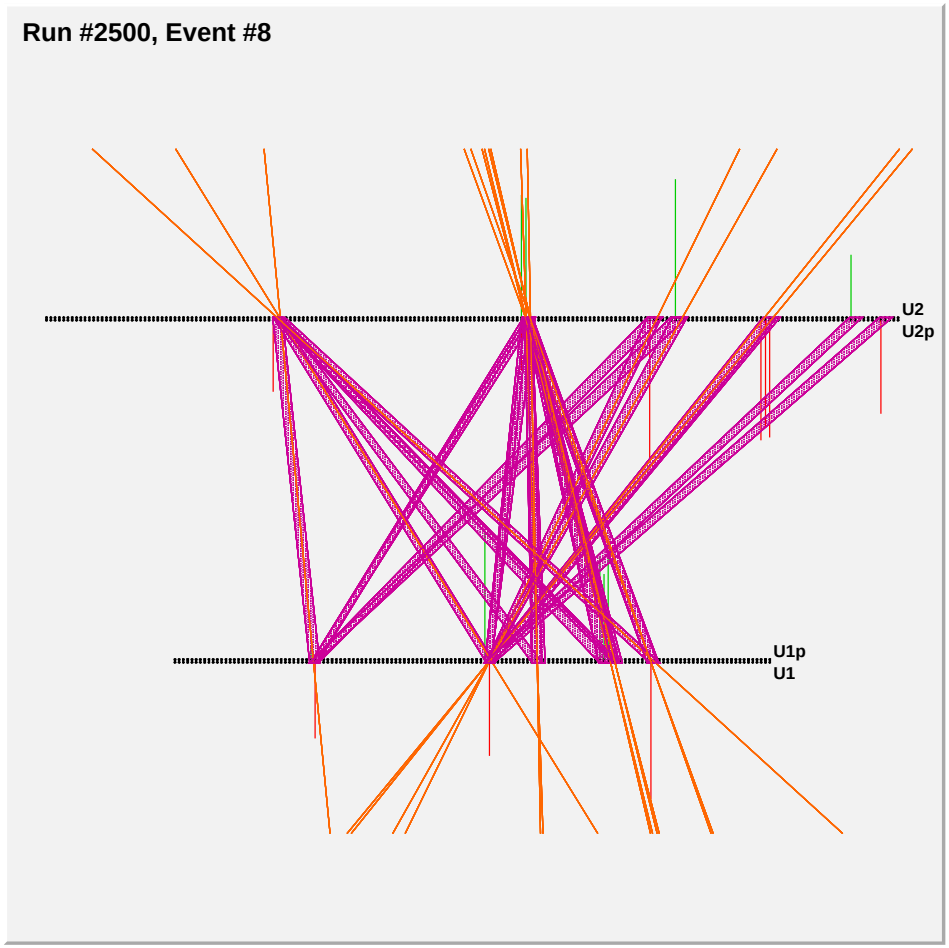
3D De-Cloning/De-Ghosting

- Build set, S, of all roads used by any 3D tracks
- Loop over 3D tracks in order of ascending χ^2
- If track's roads are all in S, accept track, remove its roads from S.
- Remove any roads from S that share any hit with the track's roads
- Quit loop if out of tracks or S no longer has roads from all projections

Event Display Planar View (E04-007)



Event Display Projection View (E04-007)



Gprof Results (E04-007, noisy data)

Printed by Ole Hansen

Jun 12, 08 0:02 **gprof−example−1k−ev.txt** Page 1/1

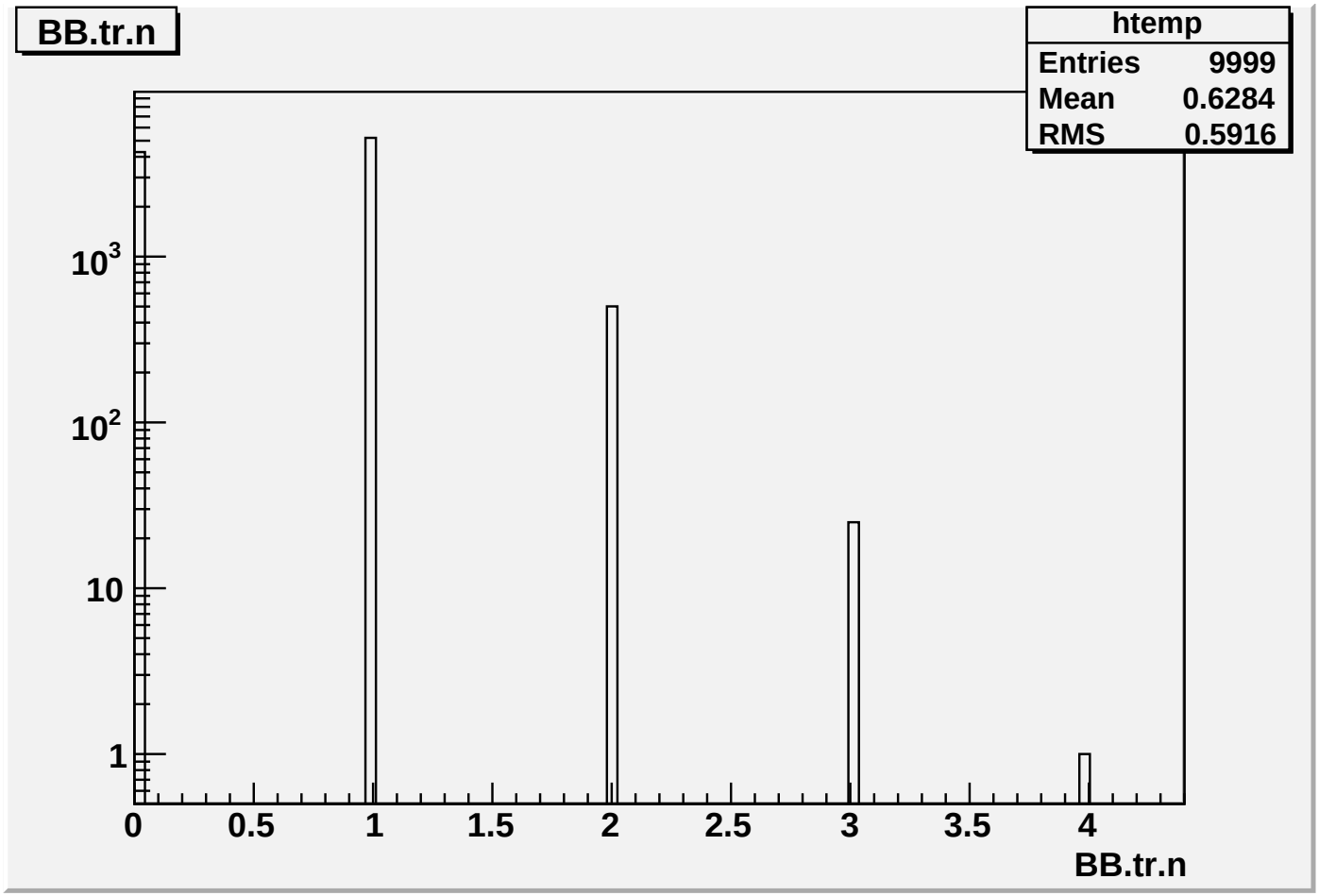
```
index % time    self  children    called     name
[1]     88.9    0.77    3.98   34001+19184037 <cycle 1 as a whole> [1]
                0.11    1.42    2948             TreeSearch::Projection::MakeRoads() <cycle 1> [2]
                0.00    1.12    2997             TreeSearch::Projection::Track() <cycle 1> [4]
                0.21    0.06  156394             TreeSearch::Road::Fit() <cycle 1> [21]
                0.07    0.17  156394             TreeSearch::Road::CollectCoordinates() <cycle 1> [24]
                0.02    0.16  570429             TBranch::Fill() <cycle 1> [35]
                0.00    0.18     999             THaAnalyzer::PhysicsAnalysis(int) <cycle 1> [36]
                0.04    0.08 2486050             THaVar::GetObjArrayLenPtr() const <cycle 1> [44]
```

Thursday June 12, 2008 gprof−example−1k−ev.txt 1/1

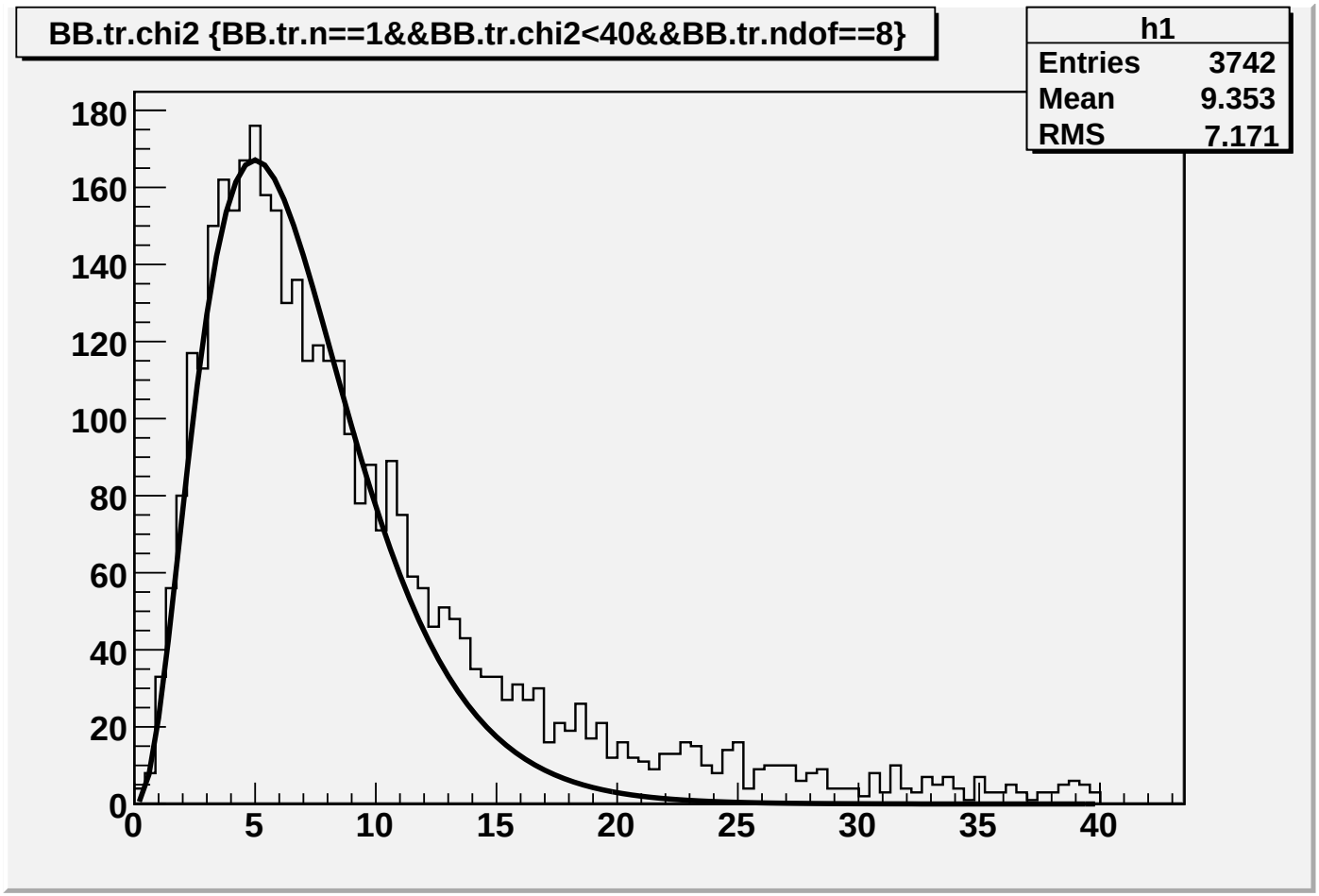
Multi-Threading of 2D Processing

- Multi-threading of `Projection::Track()` implemented (all 2D processing)
- Standard ROOT threads (pthreads)
- Maximum 3 threads (configurable)
- With noisy E04-007 data, results are disappointing (10-15% gain)
- On some older machines, multi-threaded code actually runs slower than single-threaded code! L2 cache?

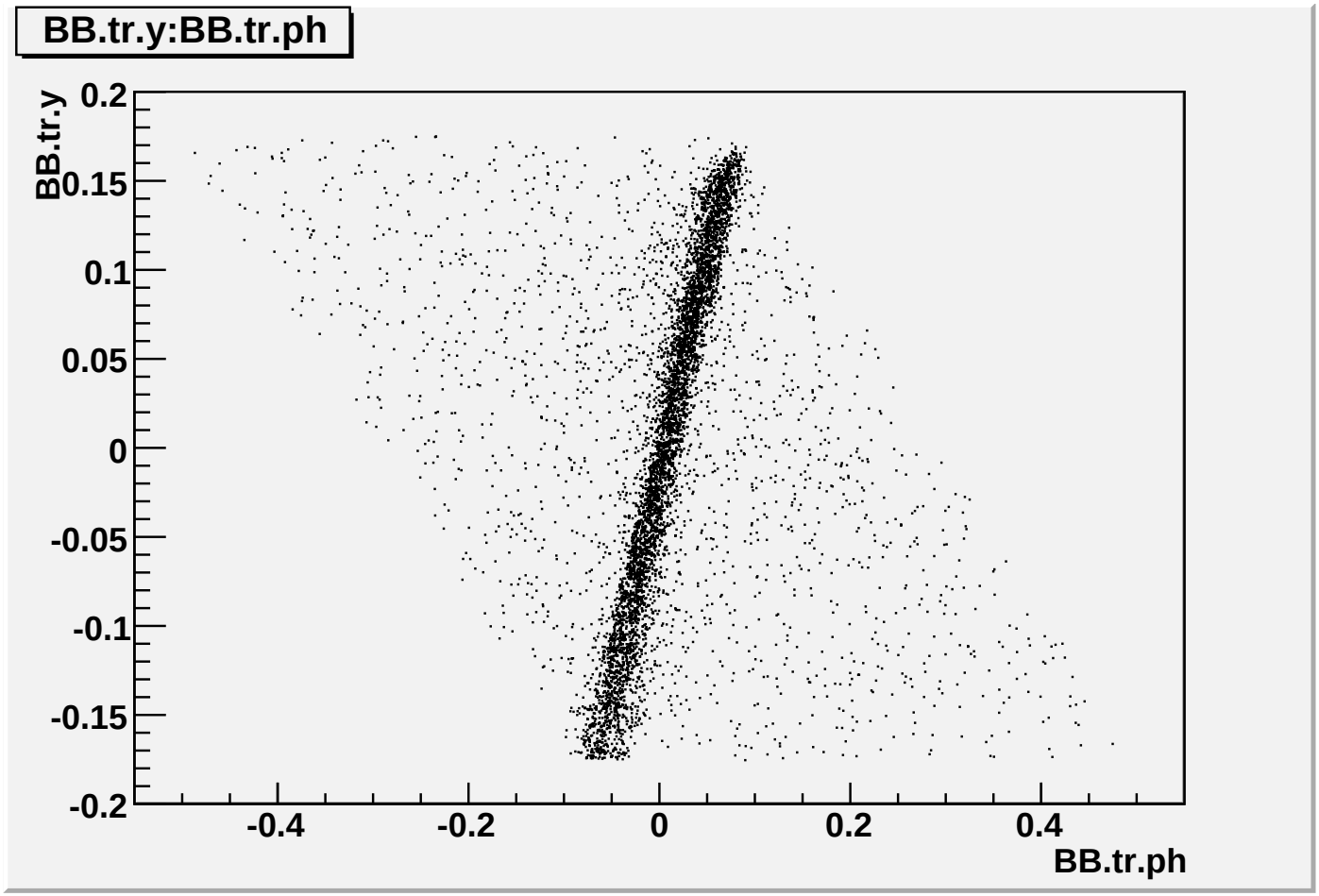
Number of tracks per event (E04-007 run 3574)



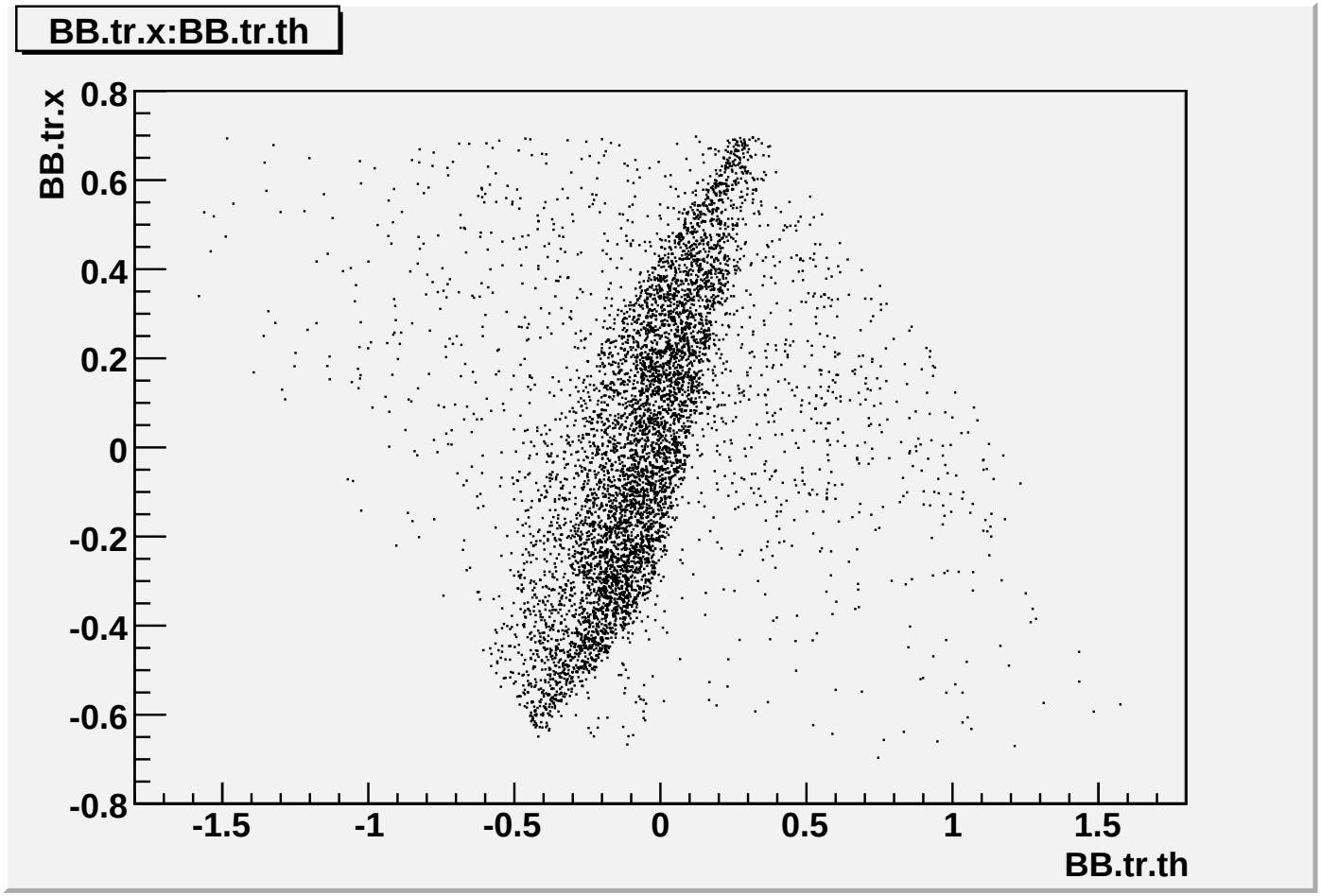
Track χ^2


BB.tr.chi2 {BB.tr.n==1&&BB.tr.chi2<40&&BB.tr.ndof==8}
h1
Entries 3742
Mean 9.353
RMS 7.171
180
160
140
120
100
80
60
40
20
0
0
5
10
15
20
25
30
35
40

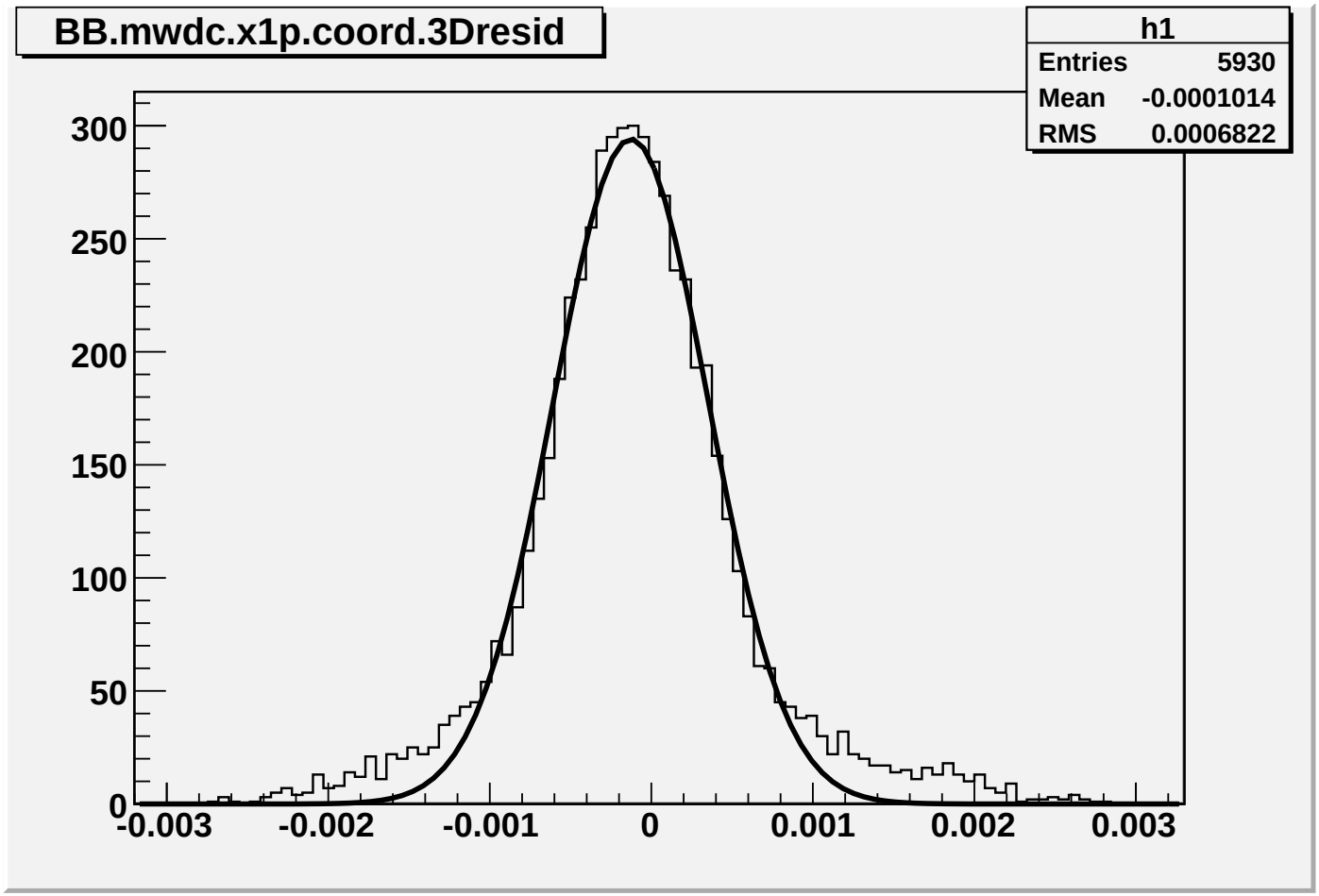
Track y vs phi (NB: approx. point target)



Track x vs theta



Hit position residuals



Experience with E04-007 data

- 12-plane configuration much slower to analyze than 15 planes. Chambers do not provide enough information to reject arbitrary combinations of front and back hits.
- Speed about 100 Hz with very noisy data on Intel Core2 6600.
- Quite sensitive to quality of geometry information
- Time-to-drift distance conversion well understood

Unfinished Tasks

- Investigate sporadic events with missing tracks
- Detailed Monte Carlo tests
- Detailed comparison with results from G_E^n code
- Optimize algorithms (eps. 2D de-cloning)
- Add ability to pre-cut using shower, scintillator, etc.
- Add $\cos\theta$, timing, fringe field corrections etc. ("FineTrack")

Volunteers welcome