Advanced Tracking Algorithms

Transversity Collaboration Meeting
20 February 2007

Ole Hansen
Jefferson Lab

Motivation

- BigBite tracking workable for G_E^n, but slow ($\mathcal{O}$(10 Hz)), memory-hungry
- Transversity vs. G_N^n: 18 planes vs. 15. Higher rates. Reconstruction may become impossibly slow.
- (Maybe) plenty of time for development if 6 month shutdown this year . . .

Types of Tracking Algorithms

Reference: R. Mankel, Rept. Prog. Phys. **67**, 553 (2004).

Global Methods

- Treat all detector hits equally
- Classic "Pattern Recognition" task
- Examples: Template Matching, Neural Networks

Local Methods

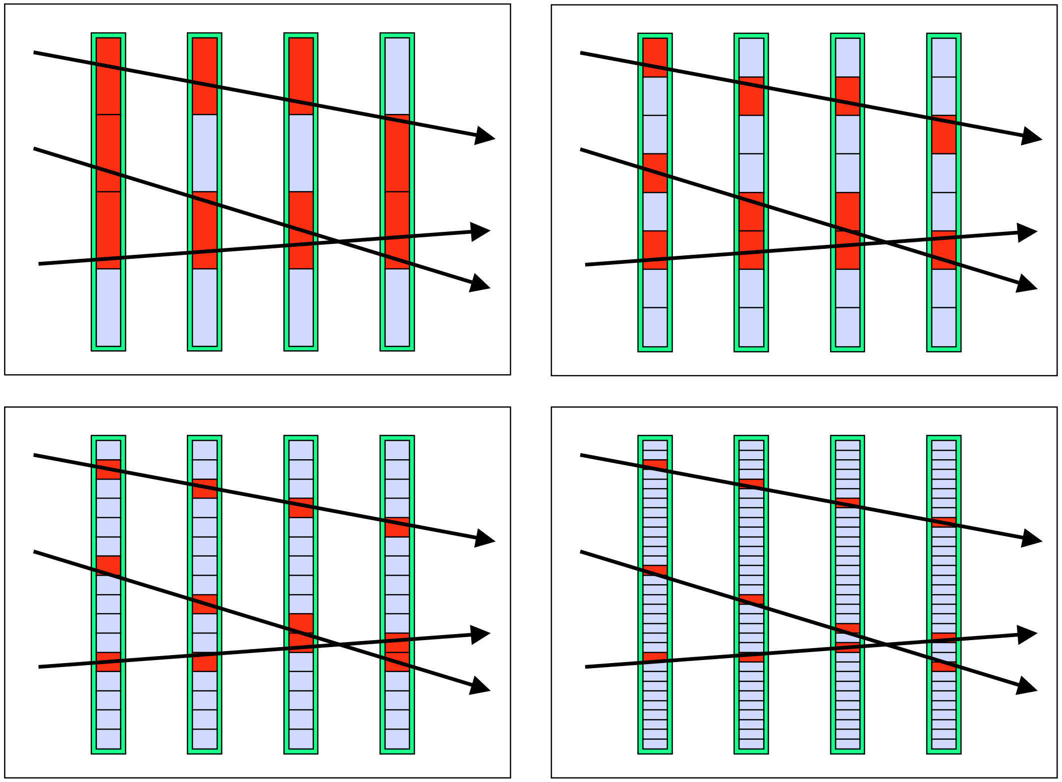
- Connect hits through detector step-by-step ("track following")
- Choose starting points ("seeds")
- Connect hits using a track model ("transport")

Global Methods: Tree Search (I)

General

- Recursive template matching with increasing resolution
- Keeps template storage requirements reasonable even for high final resolution
- Fast and efficient
- Drawbacks: requires simple geometry, straight tracks

Tree Search (II)



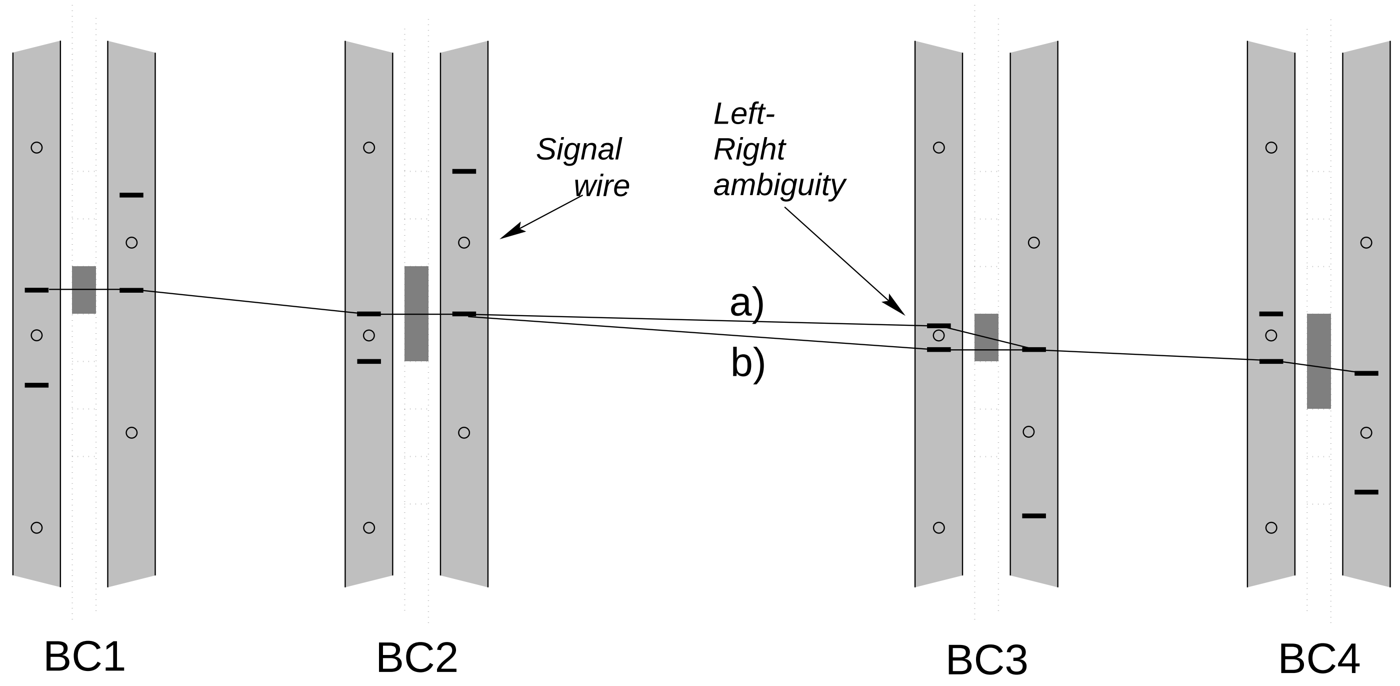
a.k.a. "track roads"

Tree Search (III)

HERMES (ref.: W. Wander, PhD thesis)

- Back chambers: similar number of wires as BigBite; similar resolution $\mathcal{O}$(1 mm); u, v, and x planes
- 126 million track patterns, reduced to 31.000 stored patterns by symmetry considerations
- Per road: $\mathcal{O}$(100) pattern comparisons, recursion depth of $\mathcal{O}$(10)
- Least-squares fit within each road to find track.
- Operate on u, v, and x projections separately, then combine in 3D

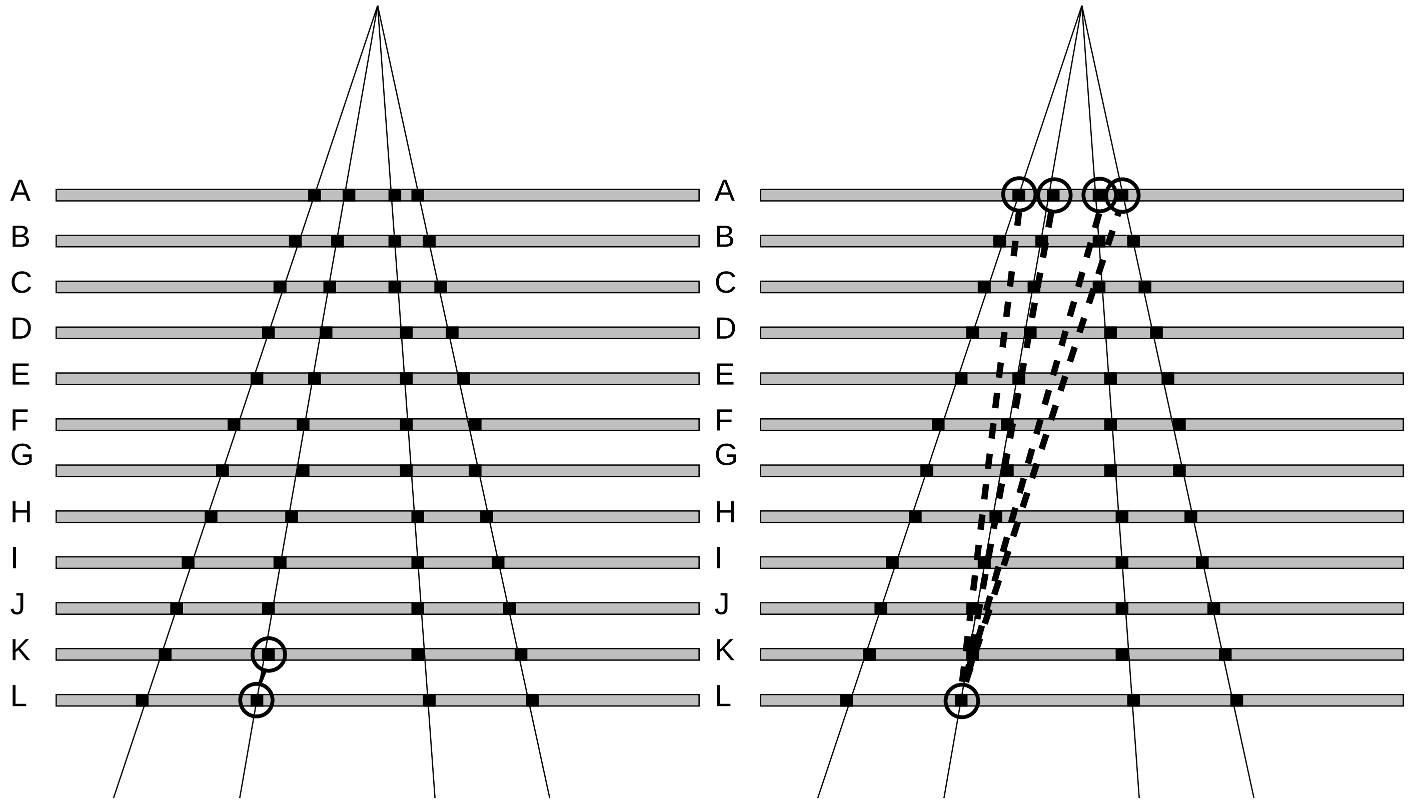
Tree Search (IV)



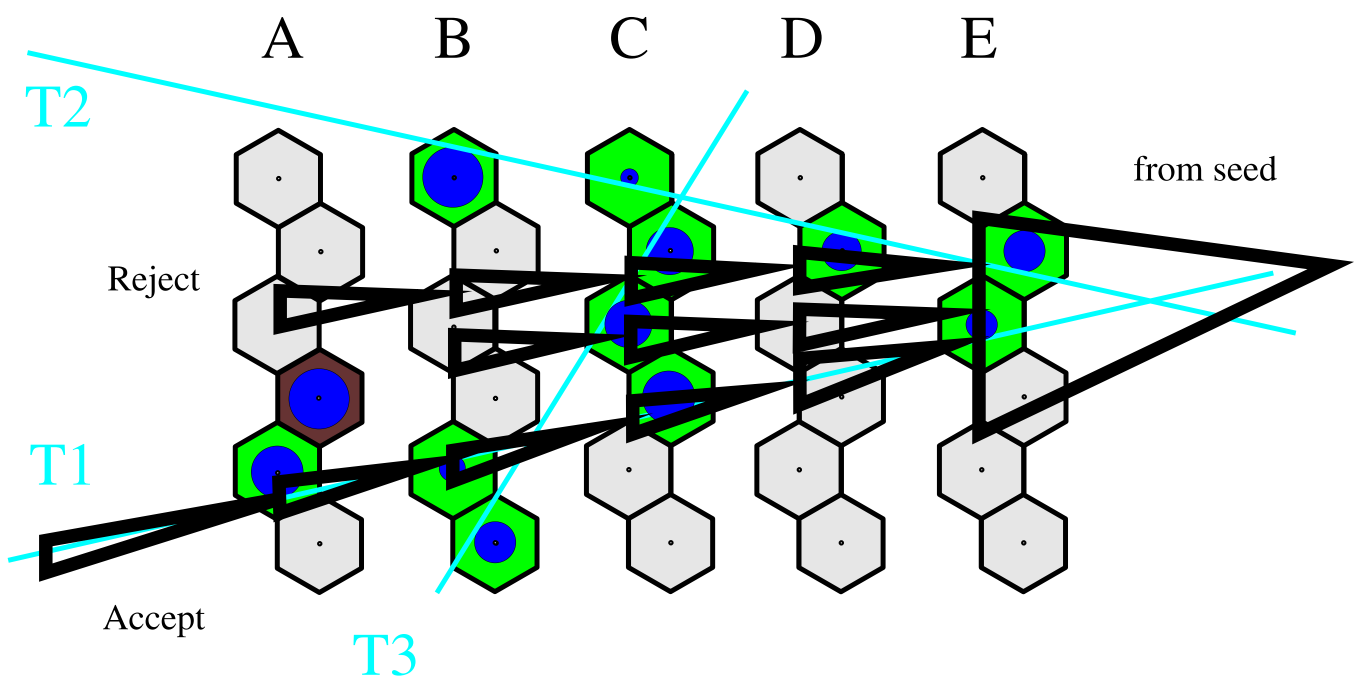
Local Methods

- Seeding required, may introduce bias
- Propagation: Kalman filter
- Naïve track following: Use best next match $\longrightarrow$ one final track, but many ways to go wrong
- Combinatorial track following: Follow all possible next hits, then eliminate candidates via quality criterion
  $\longrightarrow$ efficient, but expensive, not scalable
- Arbitrated Track Following: Allow moderate concurrency of track candidates

Seeding



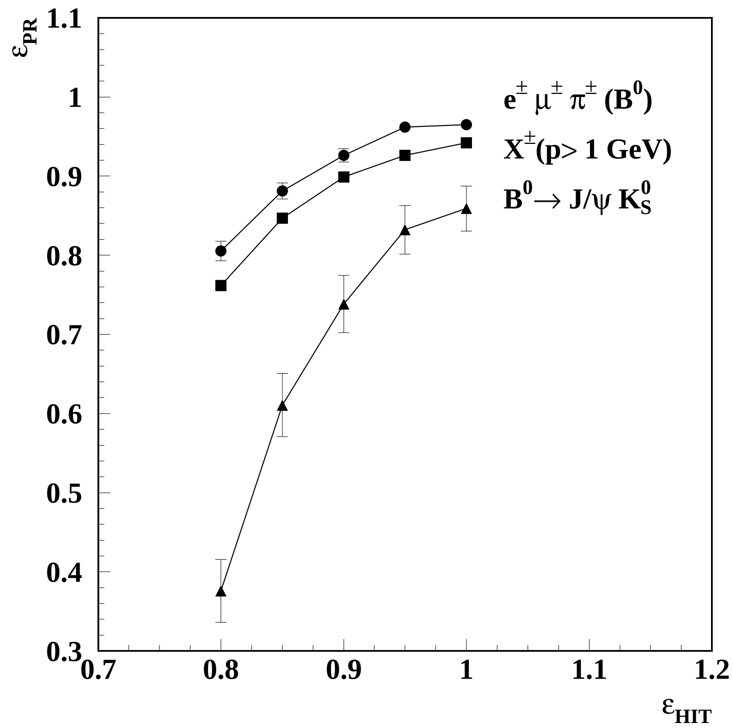
Concurrent Track Evolution (I)



Used e.g. in HERA-B main tracker

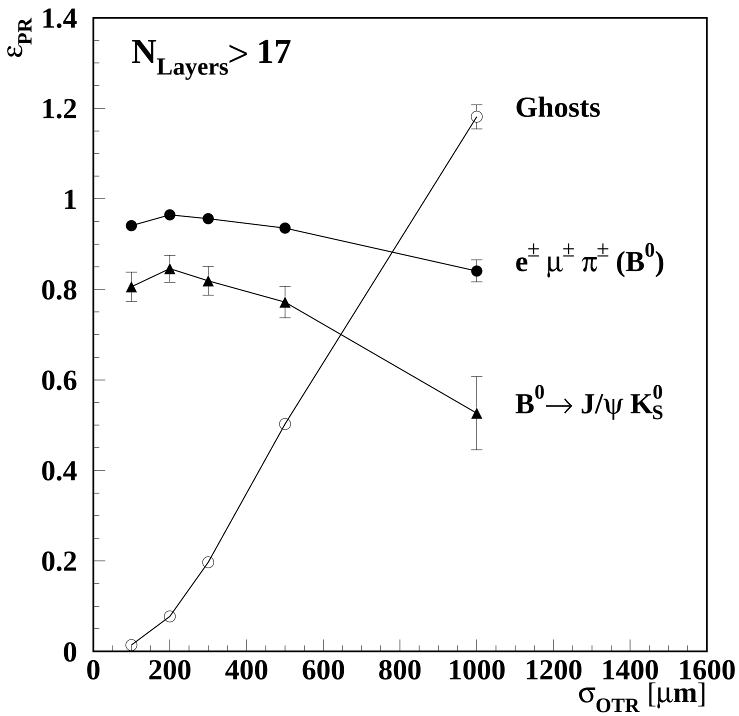
Concurrent Track Evolution (II)

Tracking efficiency vs. hit efficiency. HERA-B simulation, $N_{planes} = 24$.



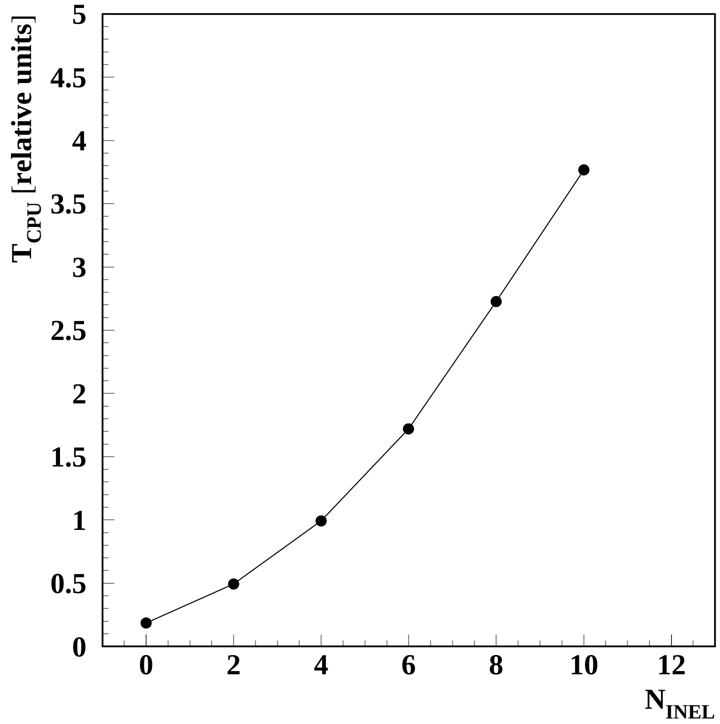
Concurrent Track Evolution (III)

Tracking efficiency vs. resolution



Concurrent Track Evolution (IV)

Mean computing time per event vs. number of tracks per event



The Kalman Filter

- Progressive least-squares fit
- Two steps: prediction and filter
- Prediction: rejects noise efficiently
- Filter: Fast; much smaller matrix to be inverted than in global fit
- Elegantly accomodates “process noise”, e.g. multiple scattering, without introducting correlated errors
- See e.g. R. Frühwirth, NIM **A262**, 444 (1987)

Summary

- Many advanced track reconstruction methods available in the literature
- Described two examples: a global tree search method and a local arbitrated track following method
- Both methods successfully used in trackers of similar or higher complexity than BigBite MWDCs & seem suitable for BigBite
- Work in progress: Experimental tree search code.