Track Finding in "Intelligent Trackers"

R. Frühwirth

Institute of High Energy Physics
Austrian Academy of Sciences, Vienna

Third International Workshop for
Future Challenges in Tracking and Trigger Concepts
FIAS, Frankfurt, February 27–29, 2012

Outline

Outline

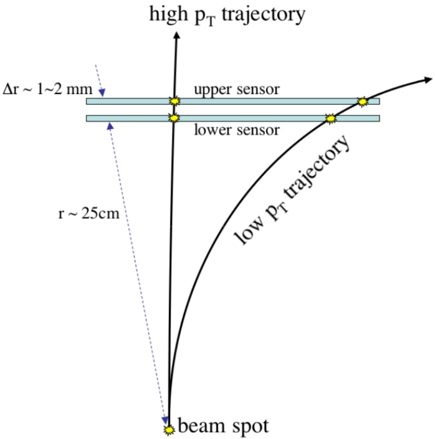
Introduction

What is an "intelligent tracker"?

- Required for phase 2 upgrade of LHC (SuperLHC)
- Trigger capability already on level 1
- Reduced readout volume by suppressing hits from soft tracks
- Needs local direction information
- Two approaches:
  - Stacked sensors: require coincidence with small deflection angle
  - Thick sensors: require small cluster size
- Stacked sensors give more precise direction information
- In combination with vertex location even p_{T} can be estimated

Introduction

An example



Introduction

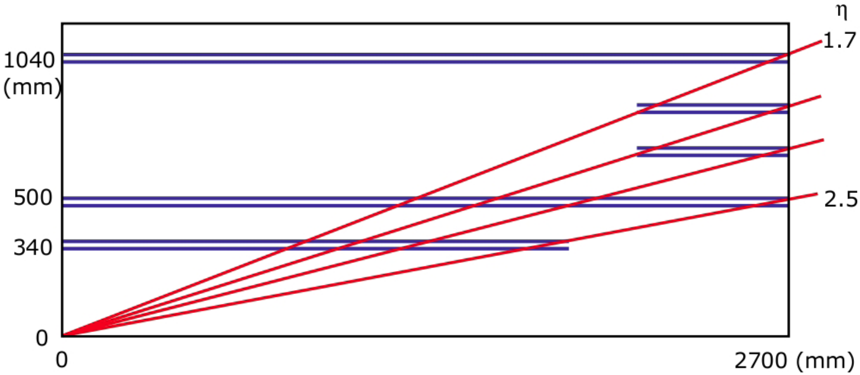
Impact on track finding

- If the stack separation is not too small, "vector hits" can be reconstructed
- Opens up new possibilities for track finding
- Have implemented a baseline version: track following with Kalman Filter
- Performance depends strongly on magnitude of stack separation in the outer layers
- Have studied various scenarios in a particular geometry
- "Long Barrel" (A. Ryd, The CMS Track Trigger Upgrade for SLHC, PoS, Vertex 2009, 040)

Outline

The "Long Barrel"

The "Long Barrel" geometry



The "Long Barrel"

Details of the "Long Barrel"

- Layers at $r = 0.30, 0.35, 0.50, 0.60, 0.95, 1.05\mathrm{m}$
- Stack separation Δr between $2\,\mathrm{mm}$ and $8\,\mathrm{mm}$
- Pixel size $100\mu\mathrm{m} \times 1\,\mathrm{mm}$
- Radiation length: $6 \times 2\% = 12\%$
- Hits are in the center of the pixel

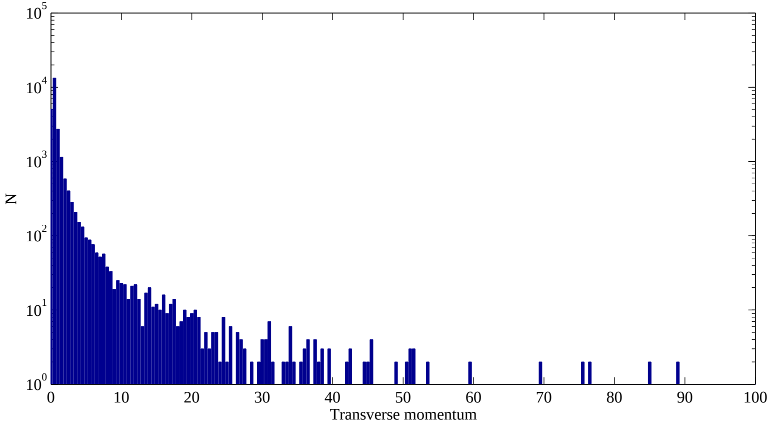
The “Long Barrel”

Simulation runs

- 500 events/run
- 1000 tracks/event
- Φ uniform in $[0, 2\pi]$
- z uniform in $[-10\,\mathrm{mm}, 10\,\mathrm{mm}]$
- η uniform in $[-1, 1]$
- $0.2\mathrm{GeV} \leq p_\mathrm{T} \leq 100\mathrm{GeV}$
- Stack separation
  - Run A: $6 \times 2\,\mathrm{mm}$
  - Run B: $2 \times 2\,\mathrm{mm}, 2 \times 3\,\mathrm{mm}, 2 \times 4\,\mathrm{mm}$
  - Run C: $2 \times 2\,\mathrm{mm}, 2 \times 4\,\mathrm{mm}, 2 \times 8\,\mathrm{mm}$

The "Long Barrel"

Distribution of p_{T}



Outline

Vector hits

Basics

- Vector hits are short track segments reconstructed from two hits in stacked sensors
- A vector hit contains four track parameters plus their covariance matrix:
  - Azimuthal position angle Φ
  - Longitudinal position z
  - Polar direction angle ϑ
  - Azimuthal direction angle $\beta = \varphi - \Phi$
- Curvature κ: see below
- Vector hits are reconstructed only for tracks above a p_{T} threshold $(1\,\mathrm{GeV})$

Vector hits

Generate cuts

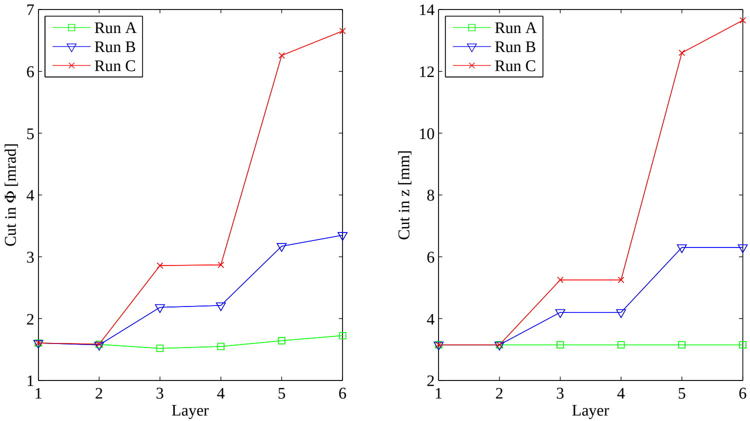
- Select tracks with $p_{\mathrm{T}} > 1\mathrm{GeV}$
- In each layer, select corresponding hit pairs
- Compute $\Delta\Phi$ and Δz
- Determine cuts by

$$c_{\Phi} = 1.05 \cdot \max |\Delta\Phi|$$
$$c_{z} = 1.05 \cdot \max |\Delta z|$$

Vector hits

Size of cuts

- Cuts increase with stack separation



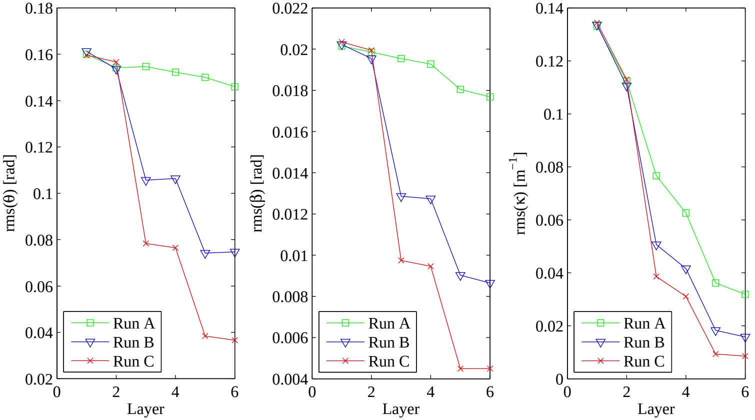
Vector hits

Vector hit reconstruction

- In each layer, select all hit pairs passing the cuts on Φ and z
- For each hit pair
  - Estimate $\vartheta = \arctan(\Delta r / \Delta z)$
  - Estimate local track direction φ and track curvature κ using two hits and the beam line (z axis)
  - Fit helix to obtain $(\Phi, z, \vartheta, \beta = \varphi - \Phi)$ plus covariance matrix
  - Curvature κ is used in the local helix track model, but retains a large error
- Resolution of Φ and z is determined mainly by the pixel size
- Resolution of ϑ, β, κ depends on the stack separation Δr

Vector hits

Resolution of vector hits



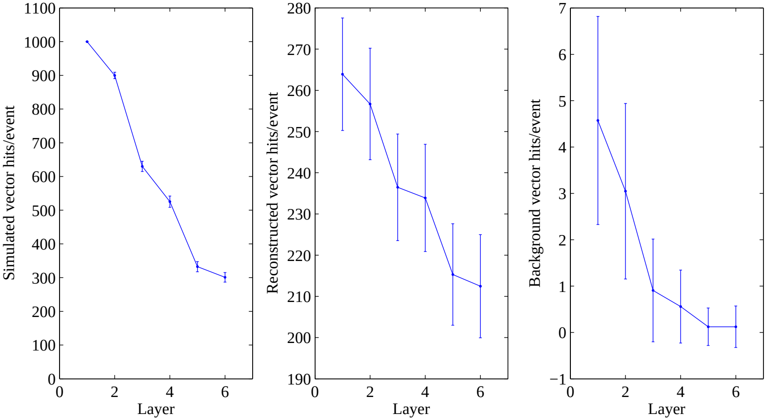
Vector hits

What about background?

- Larger stack separation implies larger cuts in Φ and z
- This might lead to more background vector hits (random combinations of hits)
- We observe only a small effect

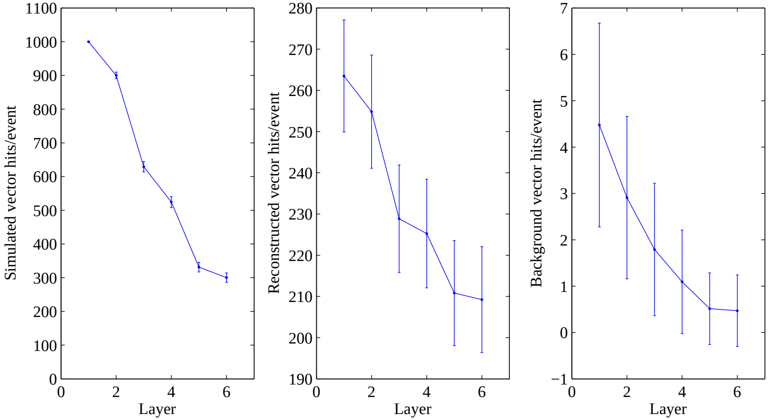
Vector hits

Vector hits Run A



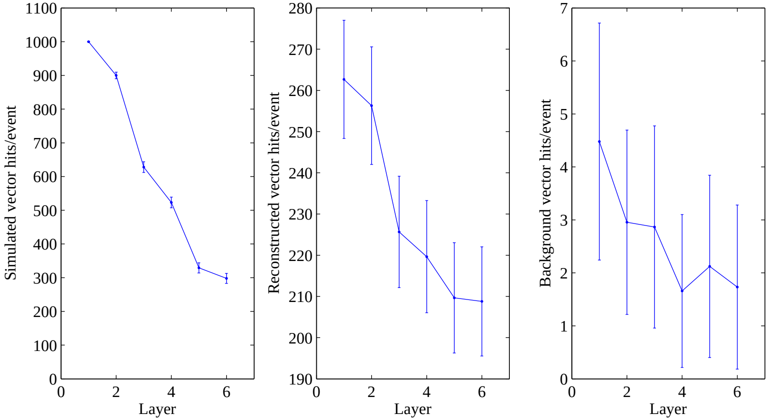
Vector hits

Vector hits Run B



Vector hits

Vector hits Run C



Outline

Track finding

Baseline version

- Track following with Kalman filter
- Start in outermost layer
- For each reconstructed vector hit
  - Extrapolate to next inner layer and define a search window ($\pm 5\sigma$ in Φ and z)
  - Compute χ^2 distance to all vector hits in the search window
  - Select closest vector hit
  - Update track state and repeat
- No inefficiencies, no combinatorics

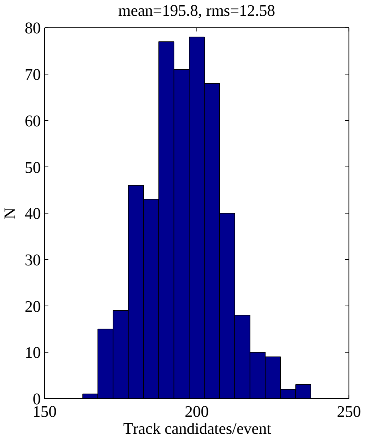
Track finding

Track analysis

- Track candidate may be
  - Unique (all hits from the same track)
  - Majority (majority of hits from the same track)
  - Ghost (no majority of hits from the same track) — this includes incomplete track candidates
- A simulated track may be
  - Found uniquely (by a unique track candidate)
  - Found in majority (by a majority track candidate)
  - Lost (not found by a unique or majority track candidate)
- In many cases a majority track candidate is really unique, because two tracks may give the same hit

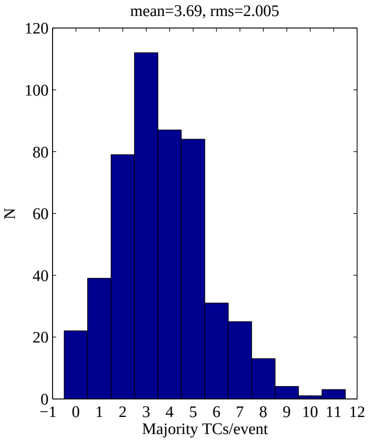
Track finding

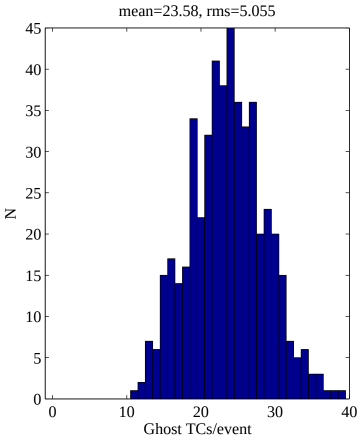
Results Run A



Track finding

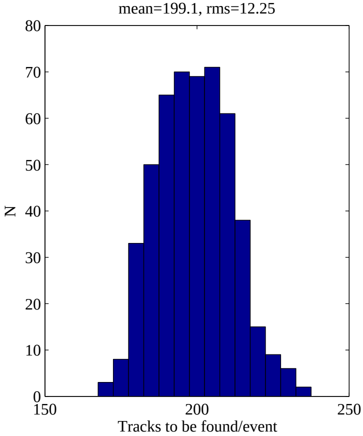
Results Run A

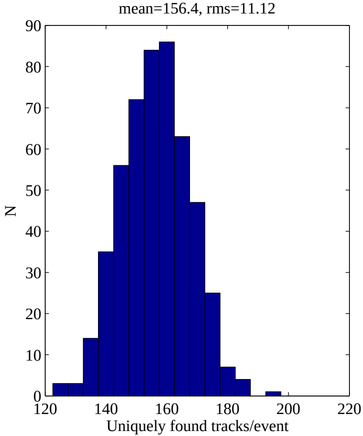




Track finding

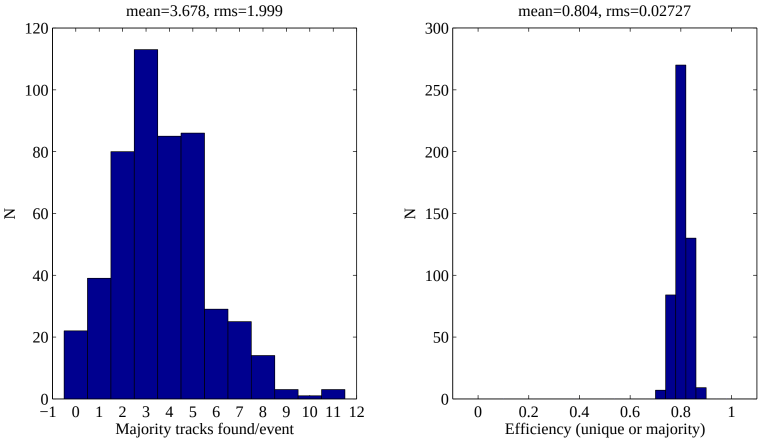
Results Run A





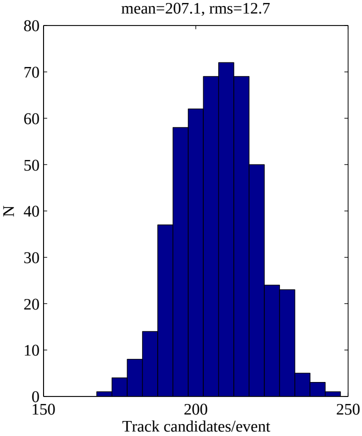
Track finding

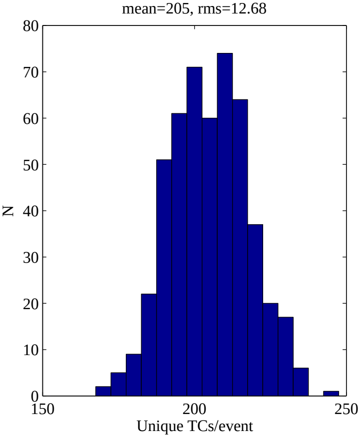
Results Run A



Track finding

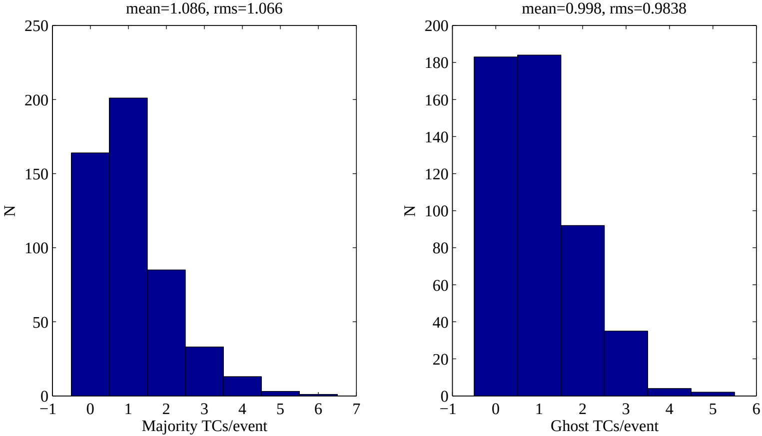
Results Run B





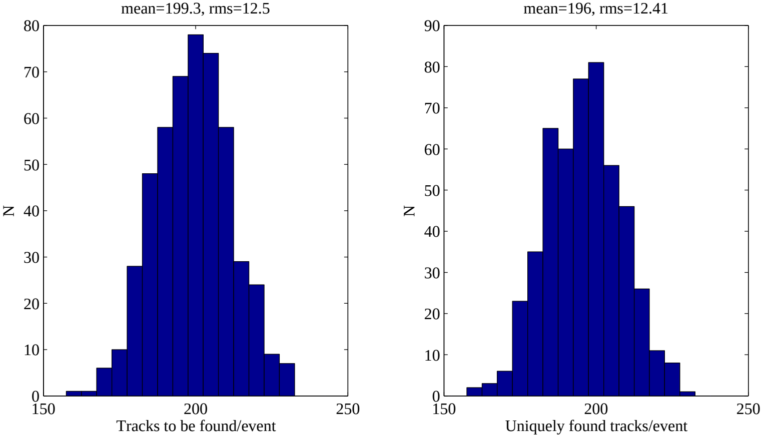
Track finding

Results Run B



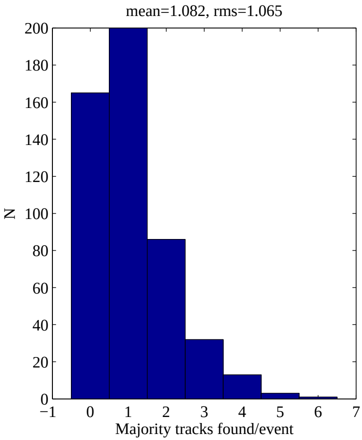
Track finding

Results Run B



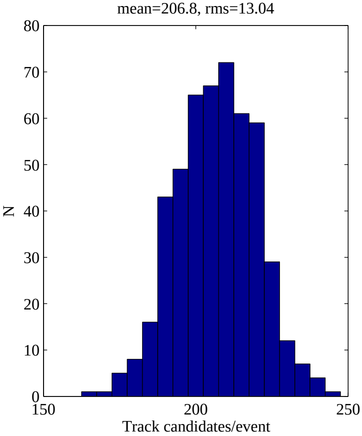
Track finding

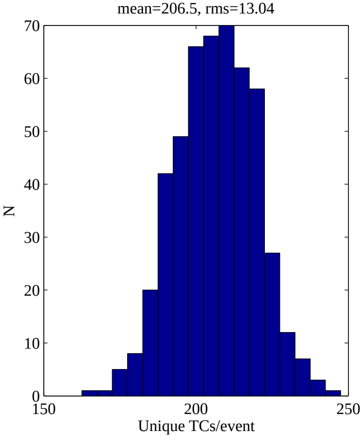
Results Run B



Track finding

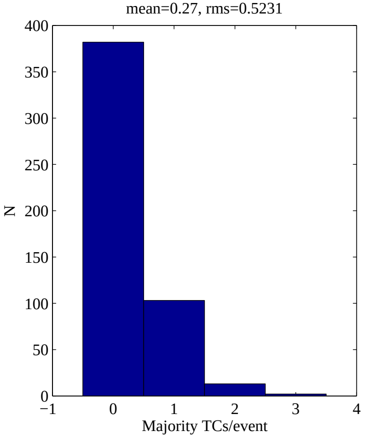
Results Run C

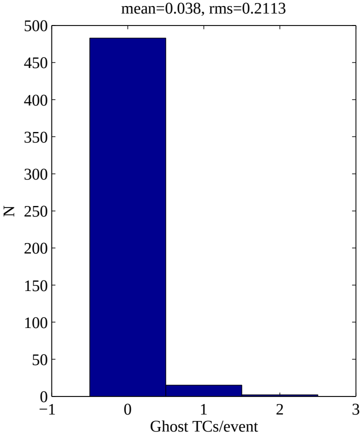




Track finding

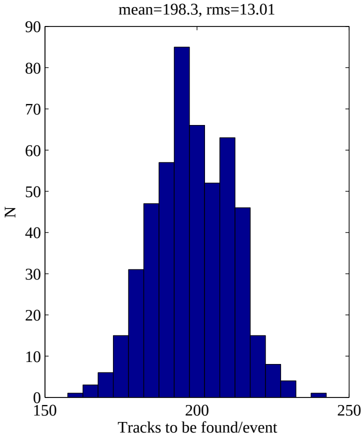
Results Run C

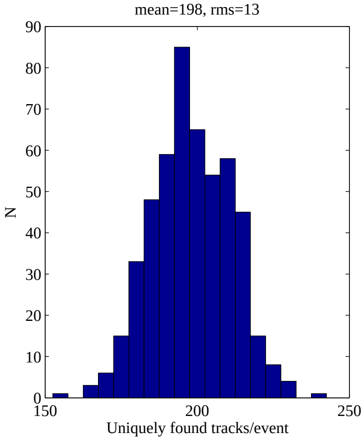




Track finding

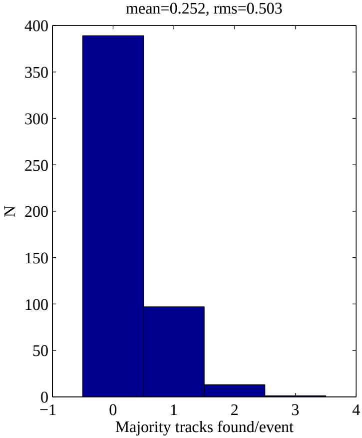
Results Run C

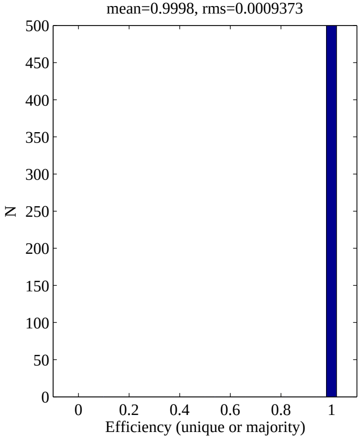




Track finding

Results Run C





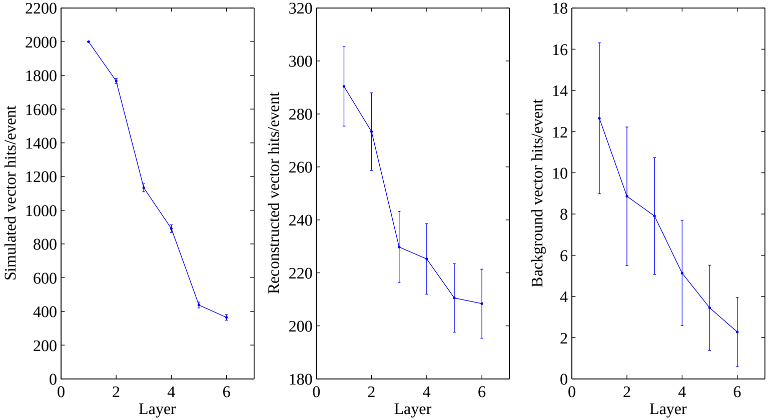
Track finding

Assessment

- Run A: poor
- Run B: very good
- Run C: perfect
- Try to make it more difficult
- Run D: add another 1000 soft tracks to each event of Run C
  - Φ uniform in $[0, 2\pi]$
  - z uniform in $[-60\,\mathrm{mm}, 60\,\mathrm{mm}]$
  - η uniform in $[-1, 1]$
  - $0.2\mathrm{GeV} \leq p_\mathrm{T} \leq 0.8\mathrm{GeV}$

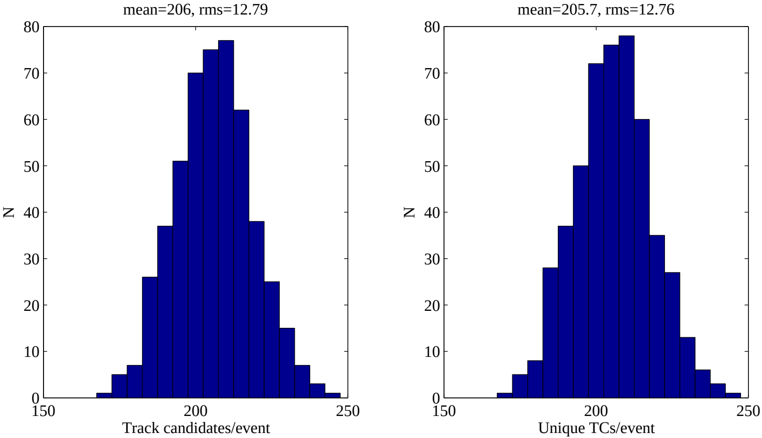
Track finding

Results Run D



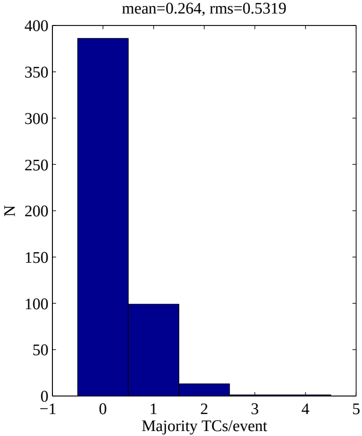
Track finding

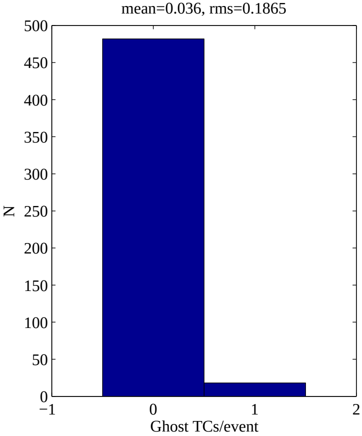
Results Run D



Track finding

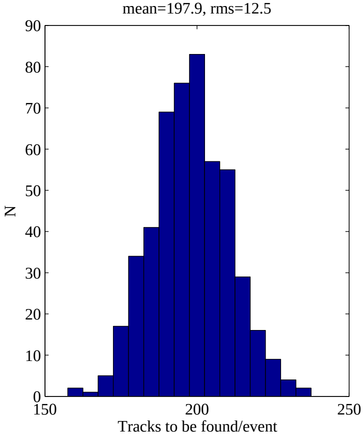
Results Run D

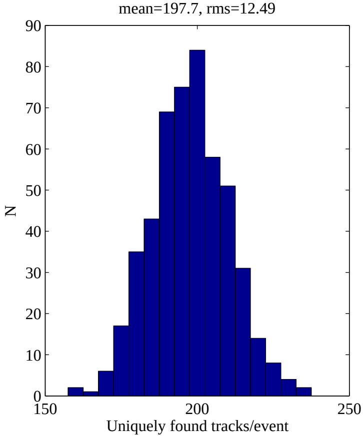




Track finding

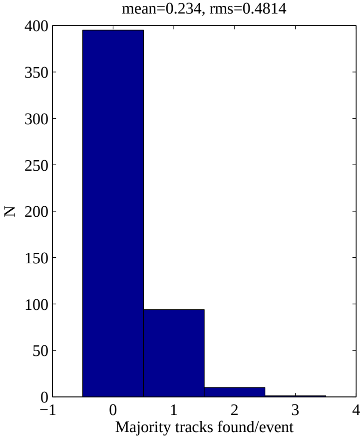
Results Run D

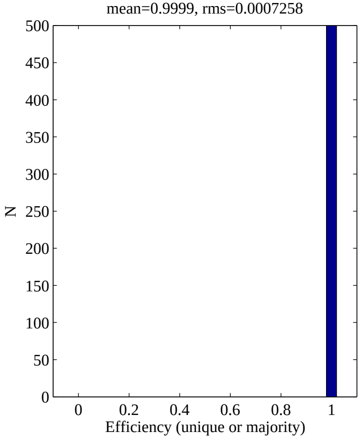




Track finding

Results Run D





Track finding

Assessment Run D

- Somewhat more noise hits
- Still perfect track finding efficiency
- Excellent baseline for further studies

Outline

Conclusions and Outlook

Conclusions

- Vector hits are useful if they have sufficiently precise direction and curvature(!) information
- If the stacks are too close, little is gained for track finding
- If the stacks are too distant, too much combinatorial background
- For track finding in real time, combinatorics should be avoided
- Have to strike balance between ease of track finding and trigger purity

Conclusions and Outlook

Input data

- Add inefficiencies
- Add more realistic noise, in particular curling tracks
- Use physical p_{T} distribution

Layout

- Optimize layer positions and stack separations in terms of track finding efficiency and momentum resolution
- Study influence of assumptions about material

Conclusions and Outlook

Algorithms

- This is just the beginning
- Full Kalman filter probably too slow for deployment in L1 trigger
- Need to develop algorithms suitable for L1
- Obvious candidates:
  - Conformal transformation plus histogramming
  - Hough transform
  - Cellular Automaton
  - Multi-layer perceptron
- Less obvious candidates?

I look forward to your comments and suggestions!

Thank you!