Track finding in silicon trackers with a small number of layers

R. Frühwirth, R. Glattauer, J. Lettenbichler, W. Mitaroff, M. Nadler

HEPHY Institut für Hochenergiephysik

Institute of High Energy Physics
Austrian Academy of Sciences

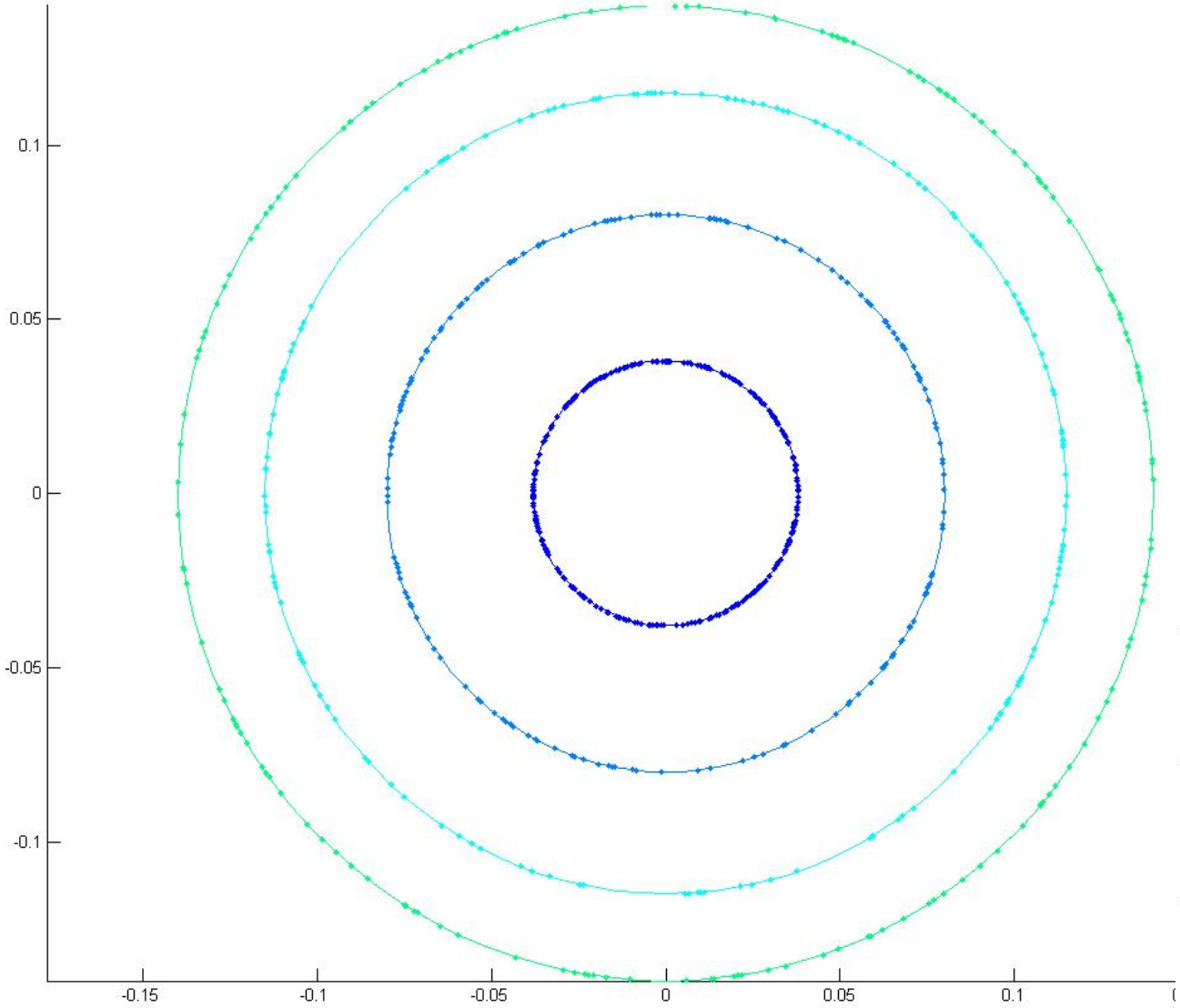
ÖAW Österreichische Akademie der Wissenschaften

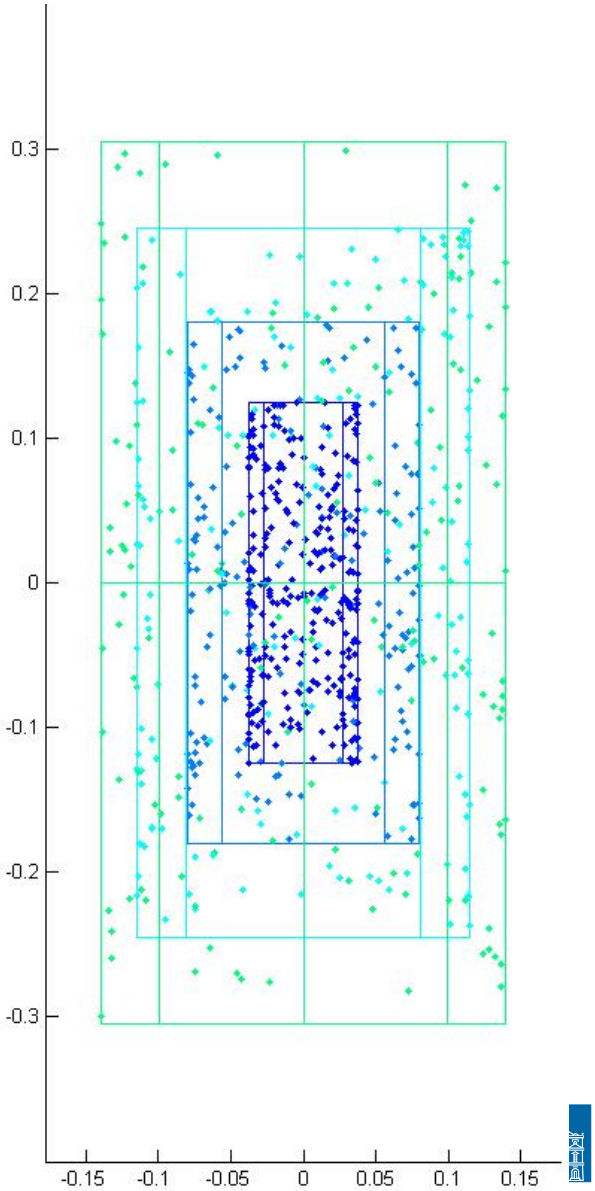
February 14th, 2013

The experiments we are working for

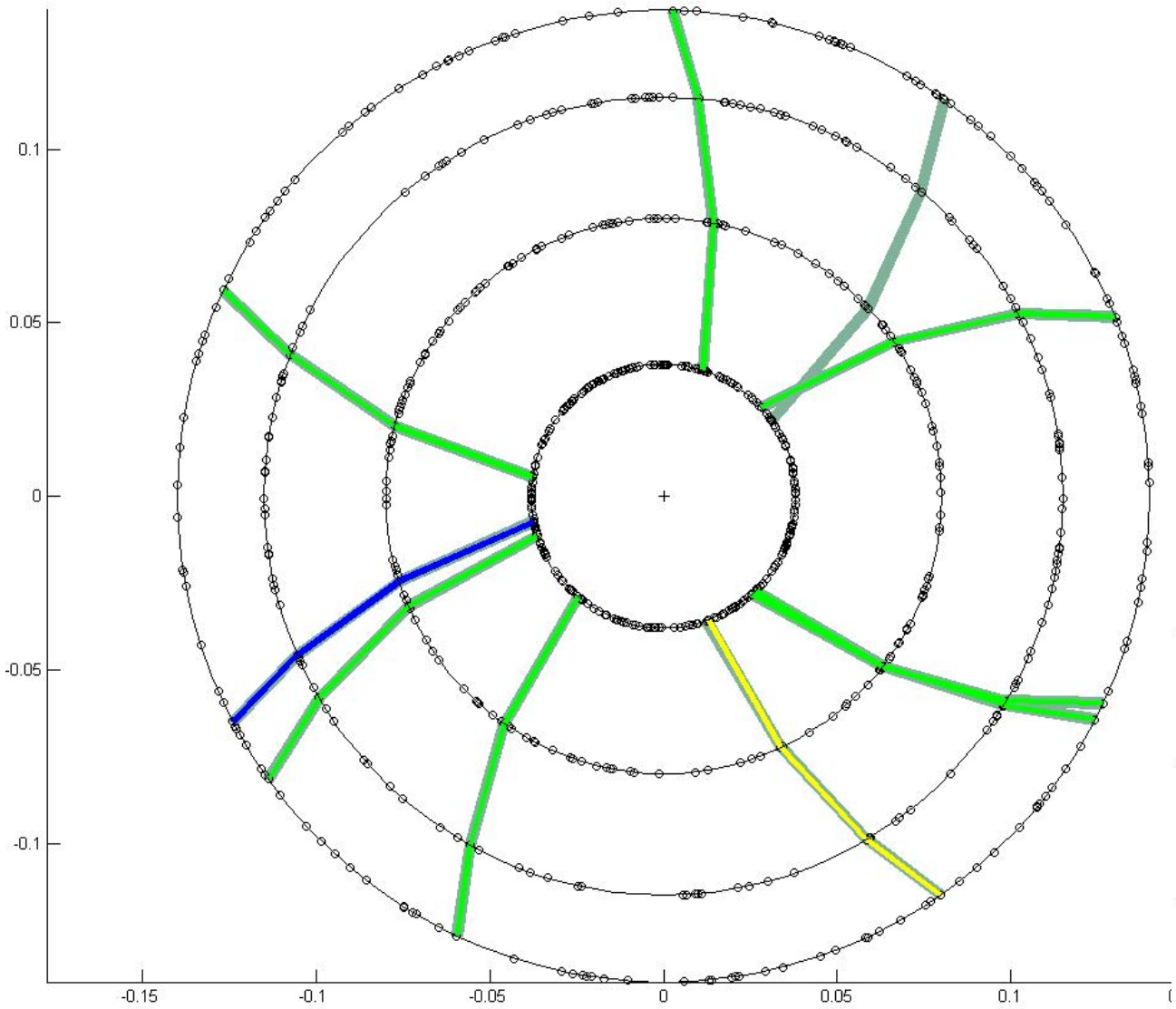
- ILD - a validated detector concept for planned International Linear Collider (ILC)
  - ILC - a linear e^-/e^+-collider with collision energy of about 500 GeV-1 TeV
  - Purpose: precision machine for measuring Higgs and BSM
- Belle 2 - the successor of Belle for the upcoming SuperKEKB collider
  - SuperKEKB - an asymmetric e^-/e^+-collider with collision energy at the $\Upsilon(4S)$ and $\Upsilon(5S)$ resonance at $\sim 10\,\text{GeV}$
  - Purpose: 2nd generation b-factory with planned integrated luminosity of $40 - 50\,\text{ab}^{-1}$ for precision measurements in the b-meson-system and BSM
- Both detectors use silicon trackers as the innermost tracking detectors

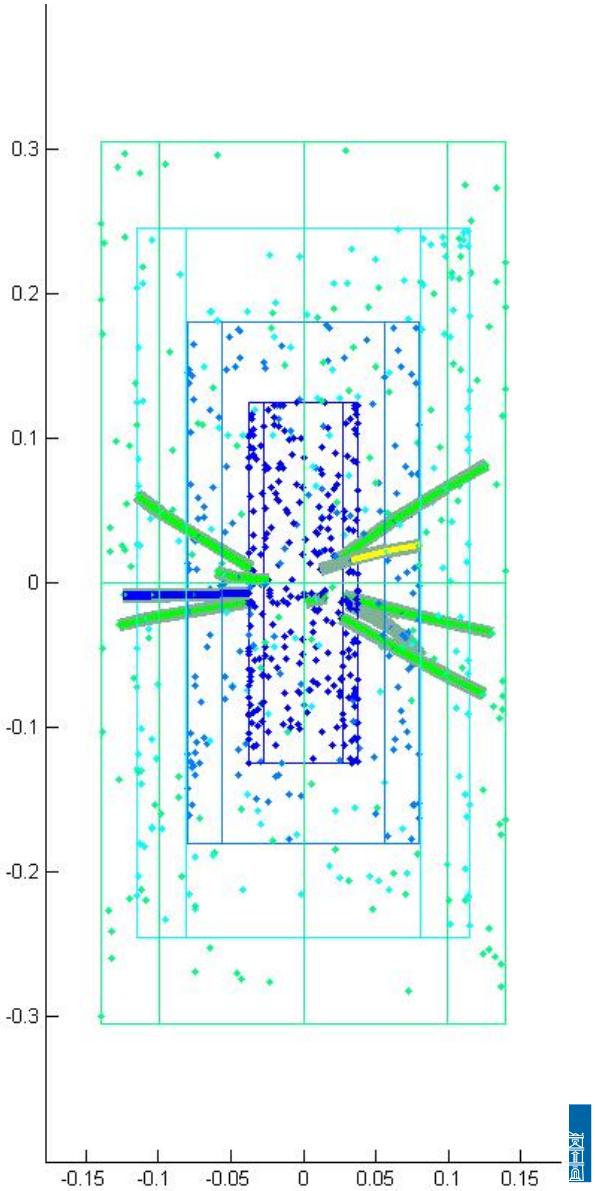
The task


0.1
0.05
0
-0.05
-0.1
-0.15
-0.1
-0.05
0
0.05
0.1


0.3
0.2
0.1
0
-0.1
-0.2
-0.3
-0.15
-0.1
-0.05
0
0.05
0.1
0.15

The solution





The difficulties

- Bad signal to noise-ratio due to machine background, ghost hits and detector noise
- The combinatorial problem is a bottle neck for reconstruction time
- Detector layout
  - In our cases a small number of layers and therefore small number of hits available for reconstruction
  - Tracking software has to consider detector specific geometry (slanted or overlapping parts, blind spots, ...)
- Detector efficiencies below 100% (missing hits) because of blind sensors, radiation damage, ...

Tracking approaches

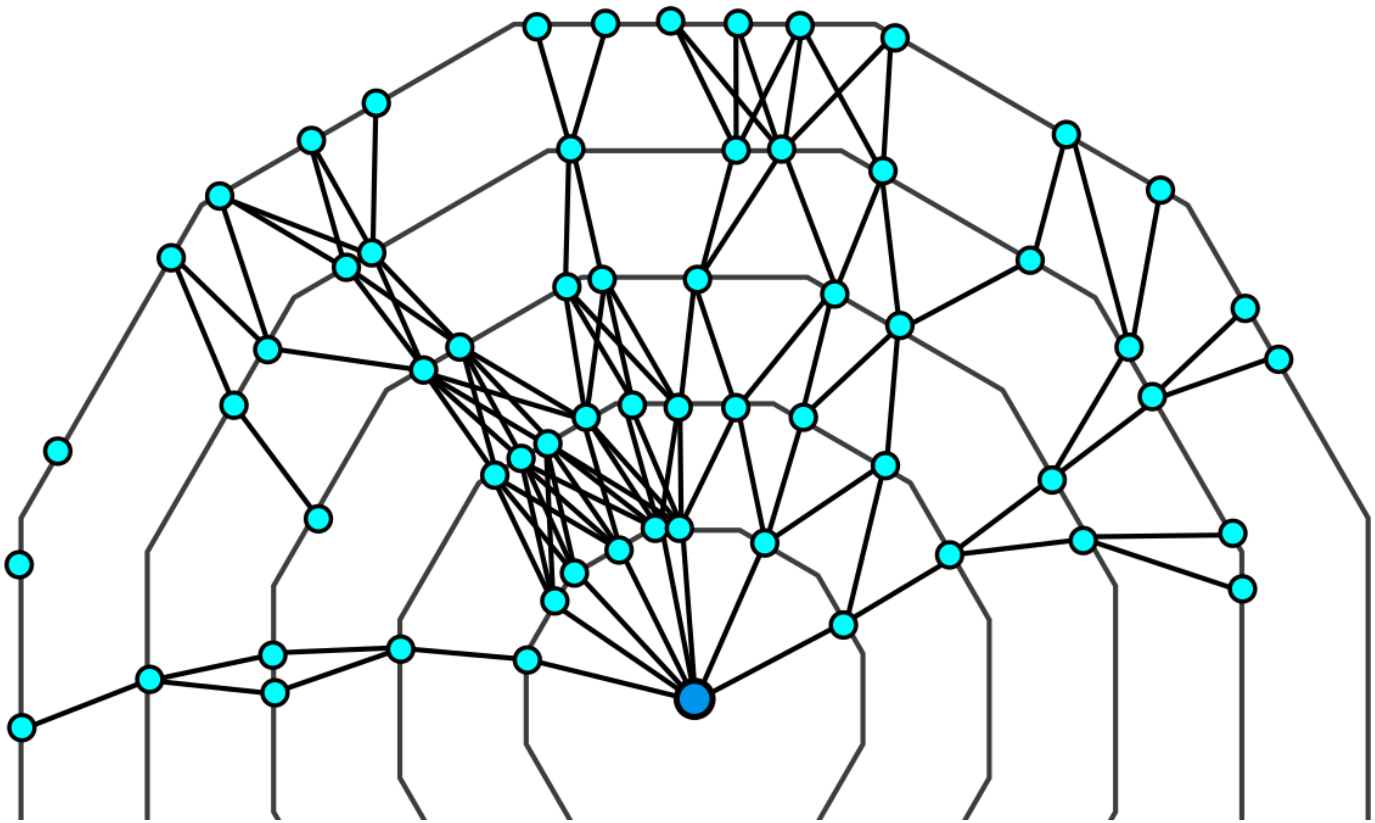
- Global
  - All hits are treated equally
  - No bias from seeding
  - Difficult for complex track models
  - E.g. Hough transformation (histogramming)
- Local
  - Use local seeds to find tracks
  - Extrapolation via track model
  - Consecutive adding of hits to the track candidate
  - Not so robust against missing hits
  - E.g. combinatorial Kalman filter

Cellular Automaton

CA in general

- Semi-global approach: all hits can be processed at the same time but CA is not so robust for missing hits
- Consists of discretized cells
- Evolves at discrete time steps
- Properties of these cells
  - Neighbourhood - each cell has got neighbours which affect each other
  - State - a value (e.g integer, boolean etc.) that can change with each iteration
  - Rules - are applied in each discrete time step and change the states

Adapting CA principle to track finding



- Cells in track finding are segments connecting 2 hits

Adapting CA principle to track finding II

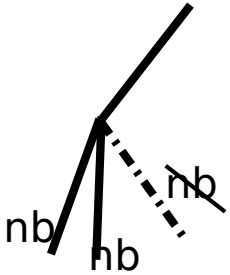
- Cells evolve depending on rules and neighbourhood
- Neighbourhood defined as: attached *inner* cells, which pass certain tests:
- These tests:
  - Have to be able to set apart genuine tracks from background
  - Should be fast
  - E.g. angles, distances, extrapolations
- States: unsigned integers starting at 0
- Rule: If *inner* neighbour has same state, cell can raise its own state by 1 at end of time step
- Steps are iterated until no cell evolves any more
- Result: state equals length of chain of compatible segments on the inside, i.e. high states indicate long chains → probable tracks

Adapting CA principle to track finding III

Cell **State** **Neighbours** **Rules**

state 0

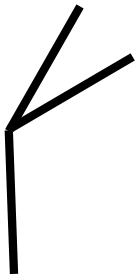
state 1



stays alive, when there are nbs with same state

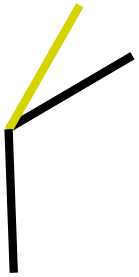
state number: number of time-steps current cell has survived so far

Initial situation



simultaneous update of all cells

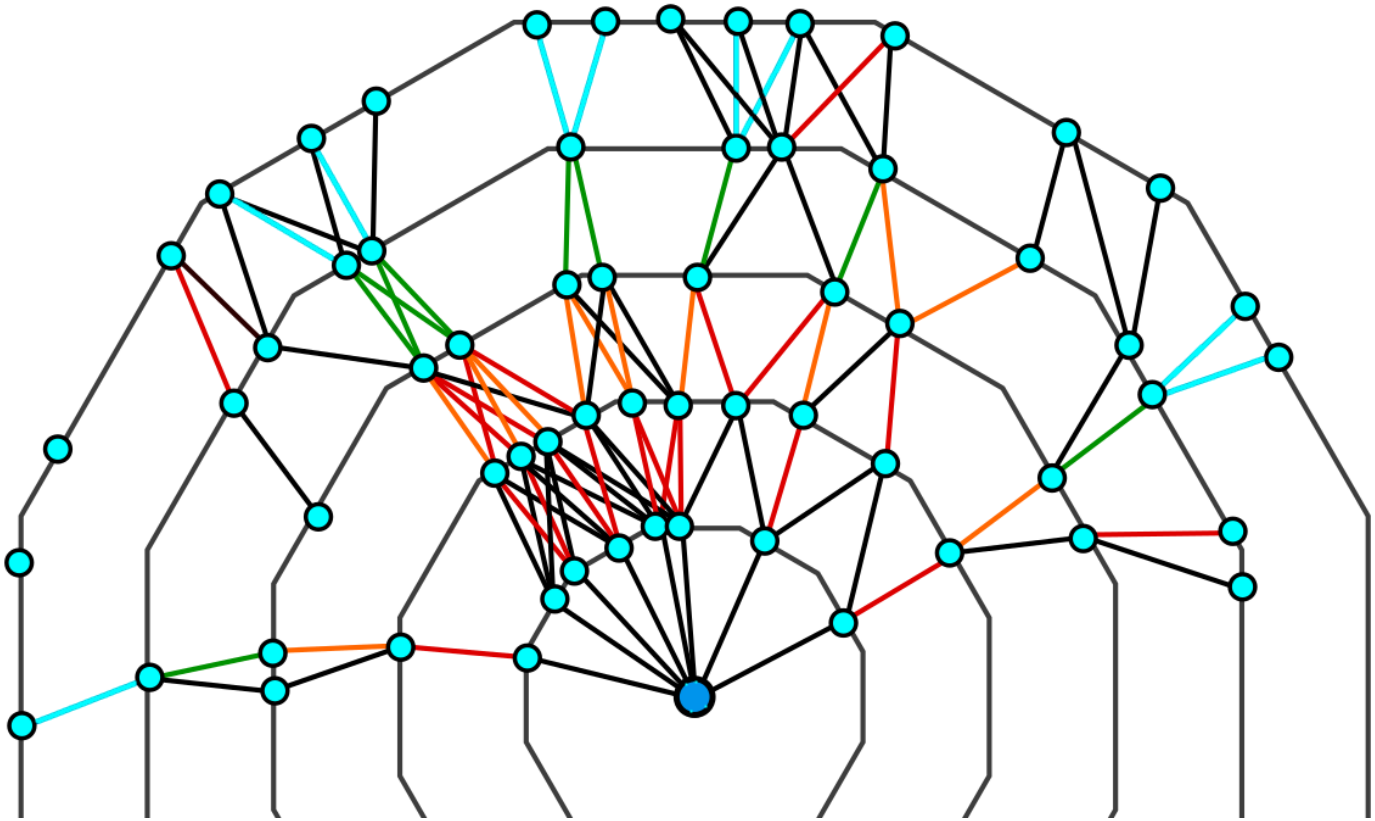
Result



repeat steps until no cell evolves anymore

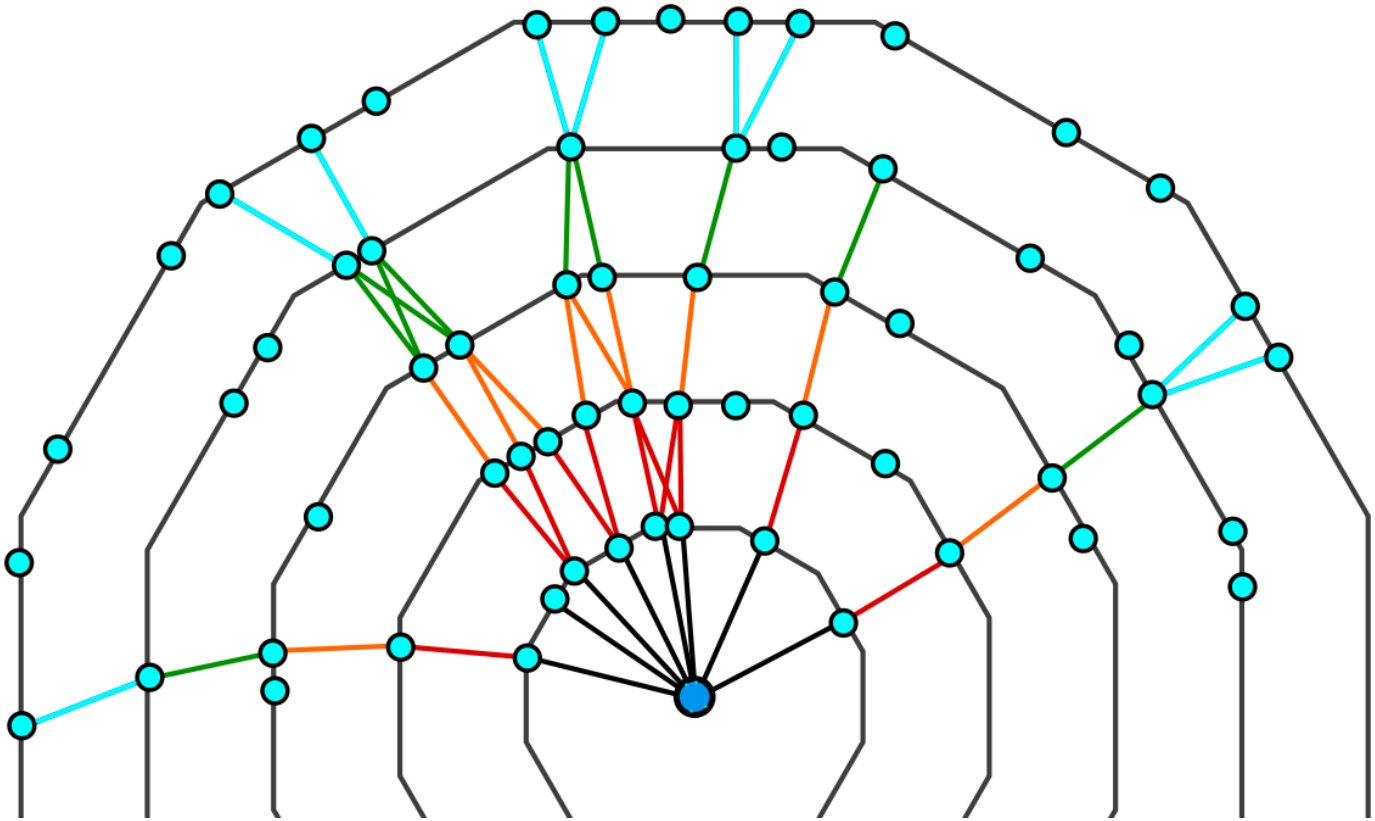
- final situation: innermost cells stay at state 0, outermost cells have got highest states

Adapting CA principle to track finding IV



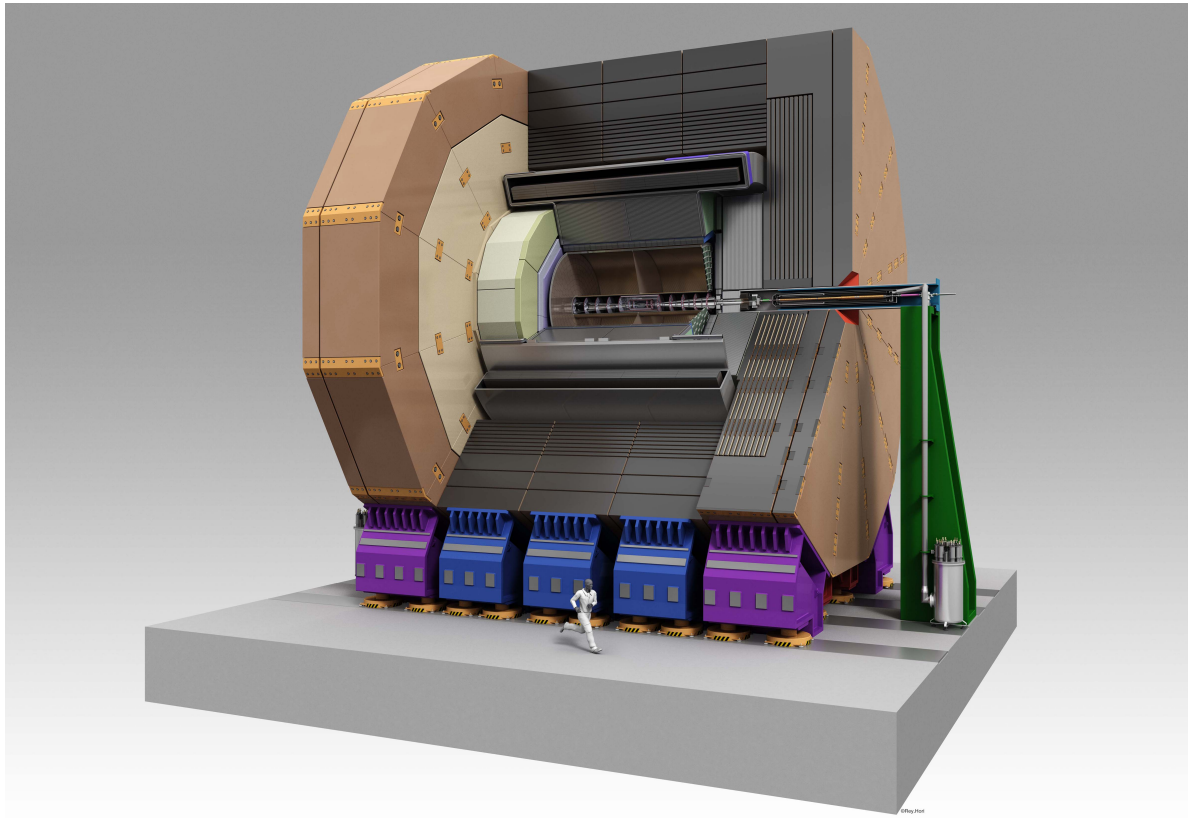
- Cells of different states:
  0 (black), 1 (red), 2 (orange), 3 (green), 4 (cyan)

Adapting CA principle to track finding IV



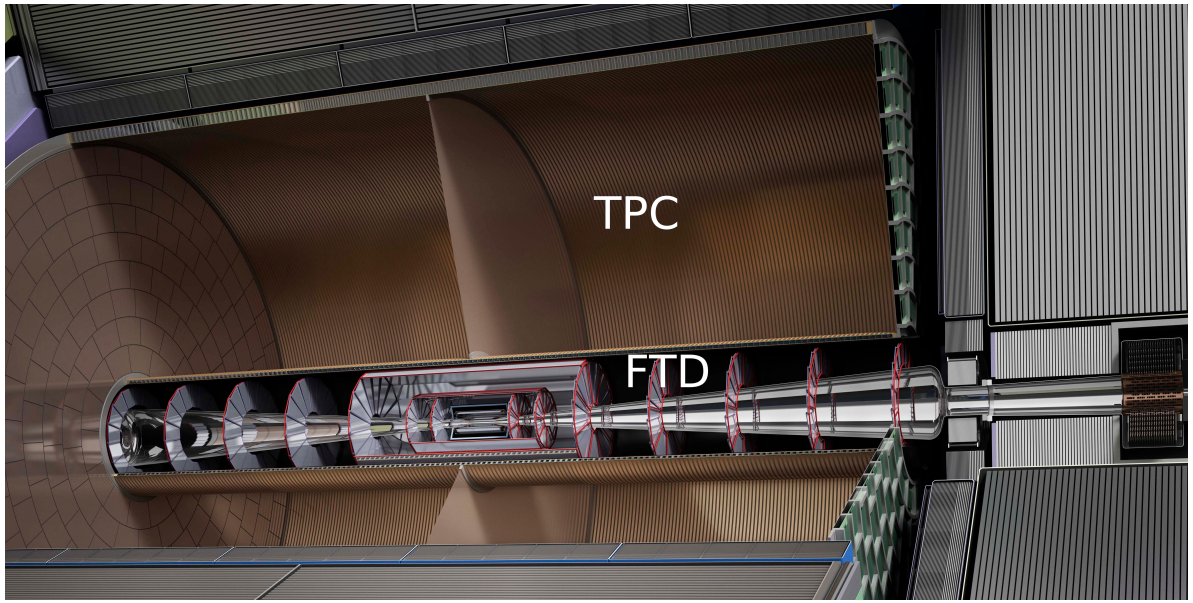
- Cells of different states:
  0 (black), 1 (red), 2 (orange), 3 (green), 4 (cyan)

The International Large Detector



- Consists of tracking + ECal / HCal + muon system
- Aims for high transverse momentum and jet energy resolutions

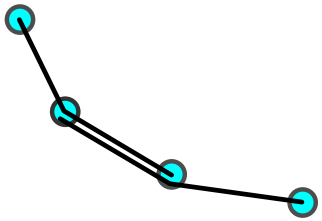
ILD tracking system



- Main tracking done by large Time Projection Chamber (TPC)
- Forward region is covered by Forward Tracking Detector (FTD) (where the Cellular Automaton is used)
  - Covers area between TPC and beam pipe
  - Grants high hermeticity
  - Two arms of 7 disk-shaped silicon detectors
  - 2 pixel disks and 5 back-to-back silicon strip detectors

Tracking situation FTD

- Large backgrounds on the inner disks (pixels), depending on used technology (integrated bunch crossings)
- Ghost hits on the outer disks (strip), especially from jets
- Cellular Automaton chosen for good background handling capability
- Concept of cells with variable lengths grants high flexibility and additional removal of fake tracks

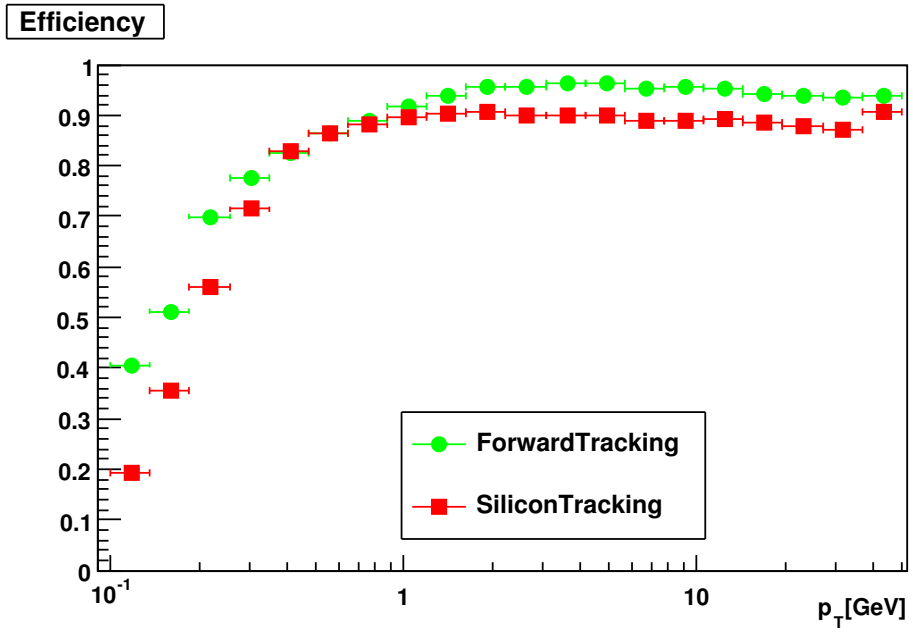


- After CA further processing with:
  - Kalman filter (estimation of track parameters)
  - Hopfield neural network (solving of incompatibilities)

Status and results

- status
  - Successfully implemented in the ILD reconstruction framework
  - Used in upcoming Detailed Baseline Design report
- results
  - Superior results in comparison to other reconstruction algorithm (based on local seeding + Kalman filter procedure), but a close race: both algorithms well suited
  - Highest gain by combination of both algorithms

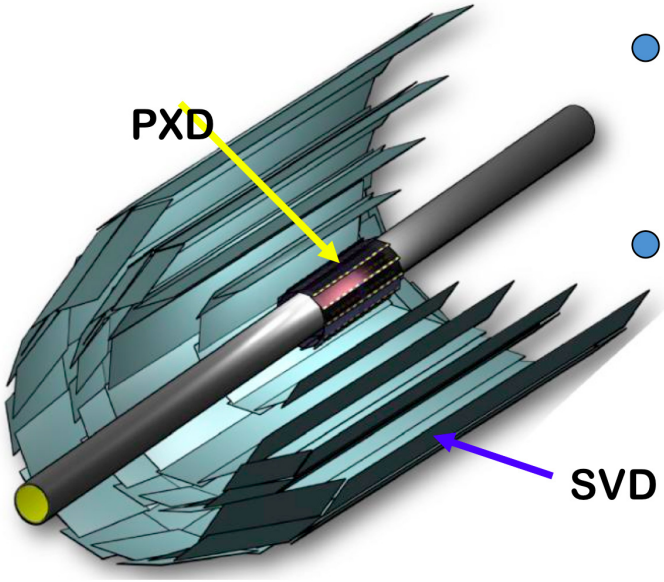
Results

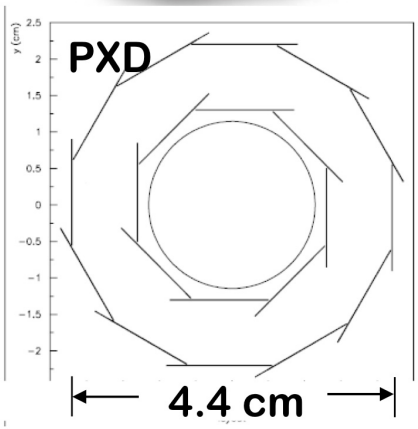


- New CA based software in green
- Efficiency with no additional background
- With background the results are quite similar (of course efficiency is less)
- Forward region is a difficult place for reconstruction (compare efficiencies well above 99% in the TPC for example)

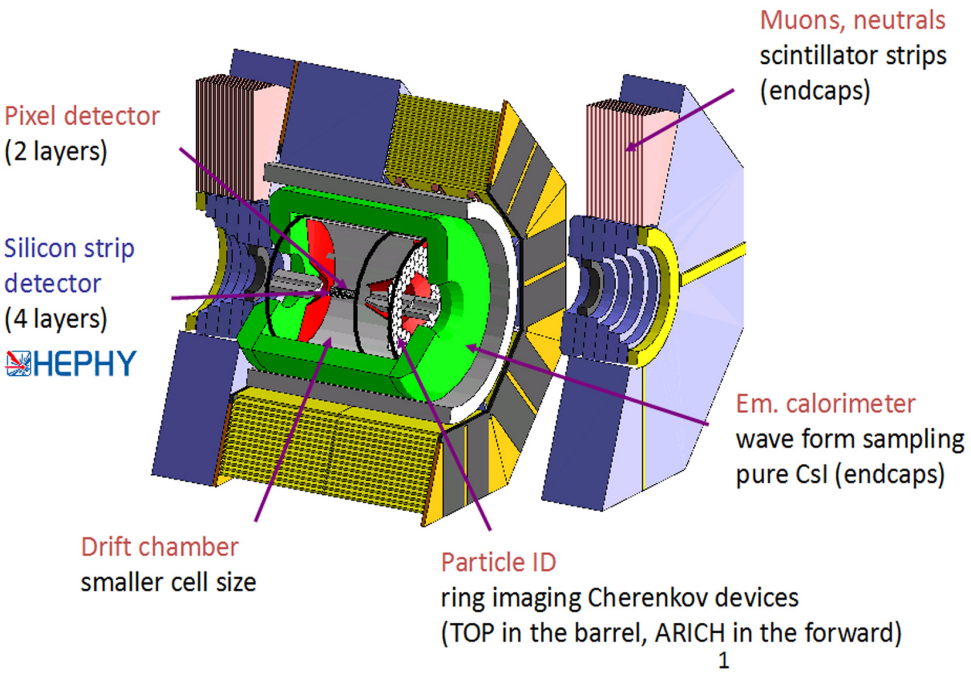
Belle II

Nano beam option: 1 cm radius of beam pipe





- 2 layer Si pixel detector (DEPFET technology) „PXD“
  (R = 1.3, 2.2 cm) monolithic sensor
  thickness 50 µm (!), pixel size ~50 x 50 µm²
- 4 layer Si strip detector (DSSD) „SVD“
  (R = 3.8, 8.0, 11.5, 14.0 cm)

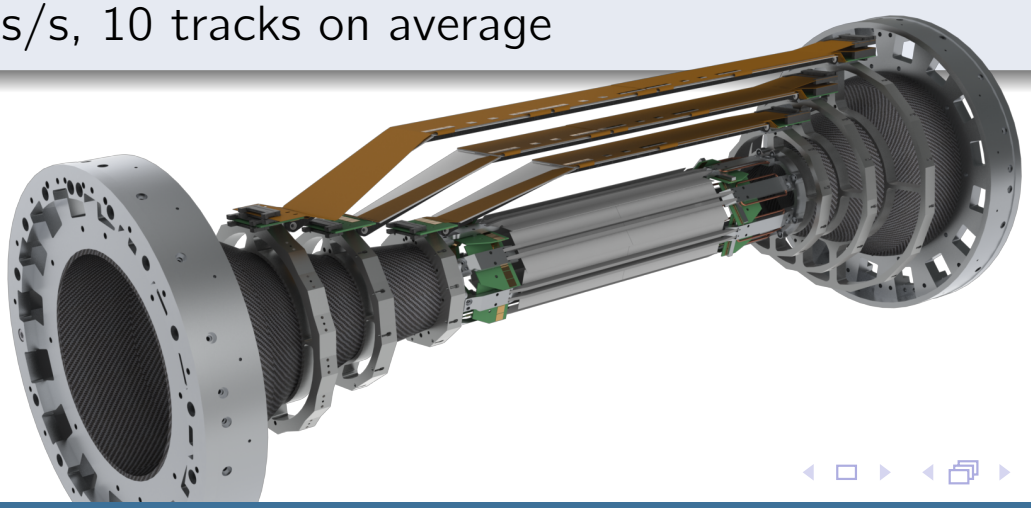


1

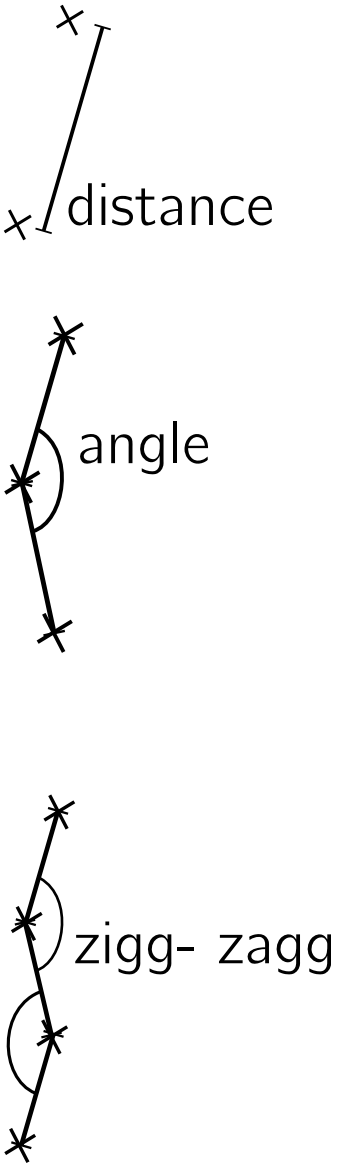
Belle 2

Important for Track Finding

- New Si detector (windmill, slanted for small θ) for Si-only track finding
  - SVD: 4 layers (double sided strips $\rightarrow$ fast (in range of ns/ROF) but ghost hits)
  - PXD: 2 layers pixel $\rightarrow$ slow (in range of µs/ROF) but no ghosts, higher resolution)
- Reconstruct low momenta ($p_T \geq 50\,\mathrm{MeV}/c$) using 3-4 layers
- Higher luminosity, 5x10^8 bunch-crossings/s
- Therefore higher background (Touschek, Bhabha scattering)
- 30k events/s, 10 tracks on average



Approach for reducing combinatorics



Schematic view of the low momentum track finder in Belle II

Unsorted hits from tracks, background, ghost coming from an event

O L Sector setup - 1-hit filter
filters by set of compatible sectors, allows momentum dependent setups

O L Segment finder - 2-hit filter
filters by distance, min&max, including virtual Segment

O L 2+1 hit filter
High occupancy bypass

O L Neighbour finder - 3-hit filter
filters by angle and Δ-distance min&max, pT

O L 3+1 hit filter
High occupancy bypass

L Cellular Automaton
evolving states, includes TC-collector

- ***Black arrows*** *represent a schematic interpretation of the possible number of combinations of hits at that point*
- ***Red arrows*** *represent high occupancy bypass strategies*
- ***Filters*** *marked with an* ***O*** *use external information generated by simulation*
- ***Steps*** *marked with an* ***L*** *cycle through several passes*

O L Post 4-hit filter
filters by zigZag, ΔpT, ...

Circle fit
High occupancy bypass

Kalman filter
Calculates QI's

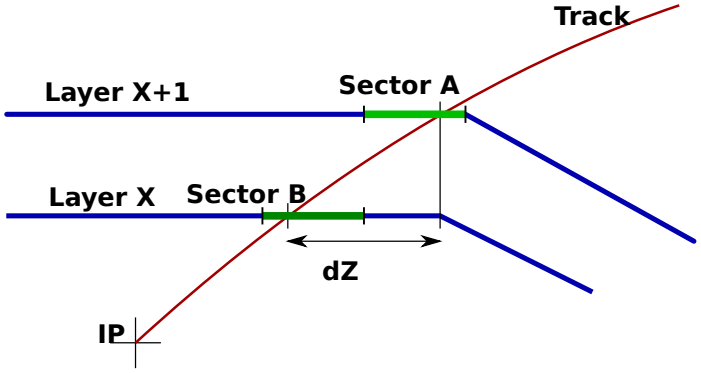
Hopfield Network
uses QI's to find best subset among overlapping TC's

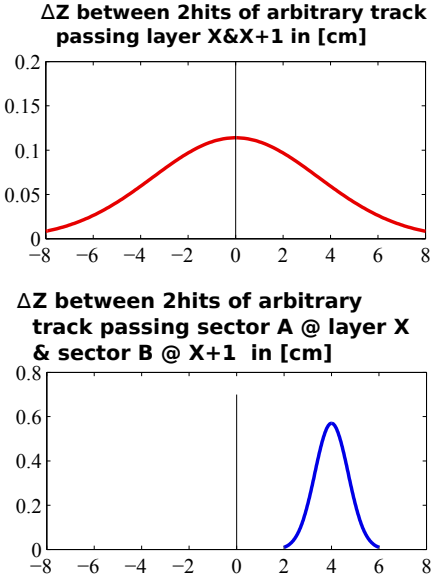
Clean TC's

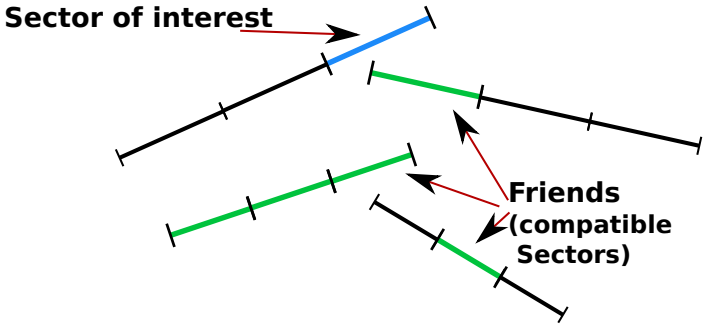
Sectors

Motivation using sectors:

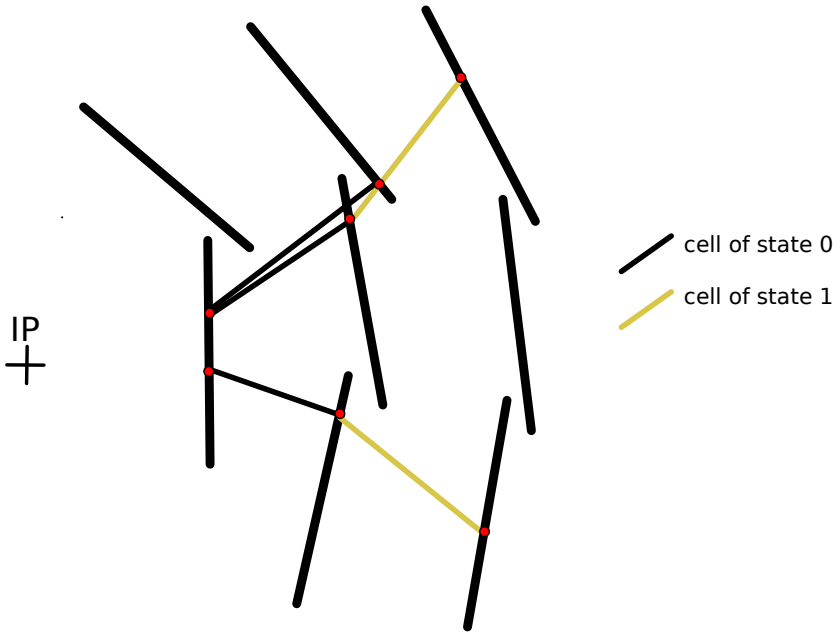
- Windmill structure and slanted sensors forbid simple layer-wise tests → at least sensor-specific tests needed
- Better: subdividing sensors in sectors and storing friend-lists
- → Allows customized filters to reduce combinatorics
- → Allows multi-pass optimizing for different momenta and curling tracks



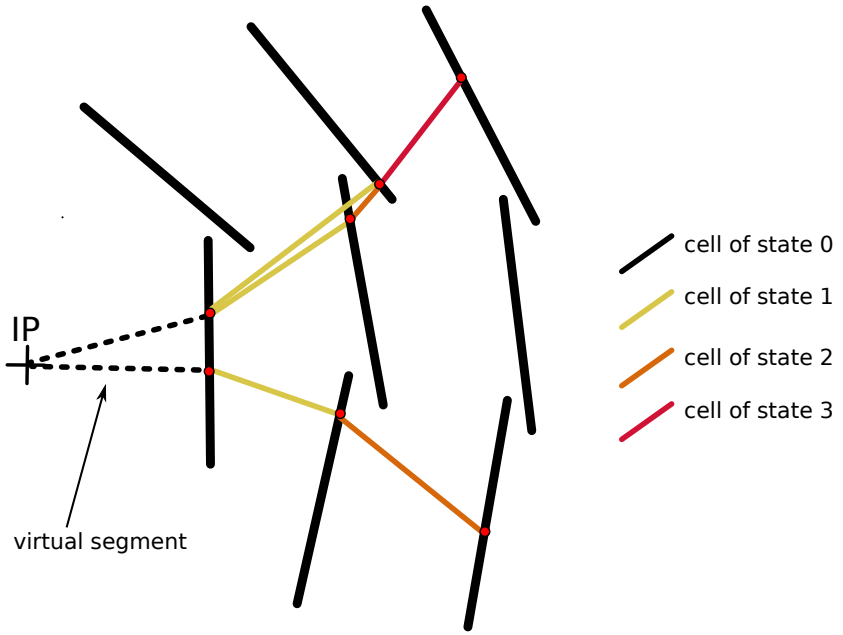




Adapting CA to 3-4 layers, virtual segment and sectors

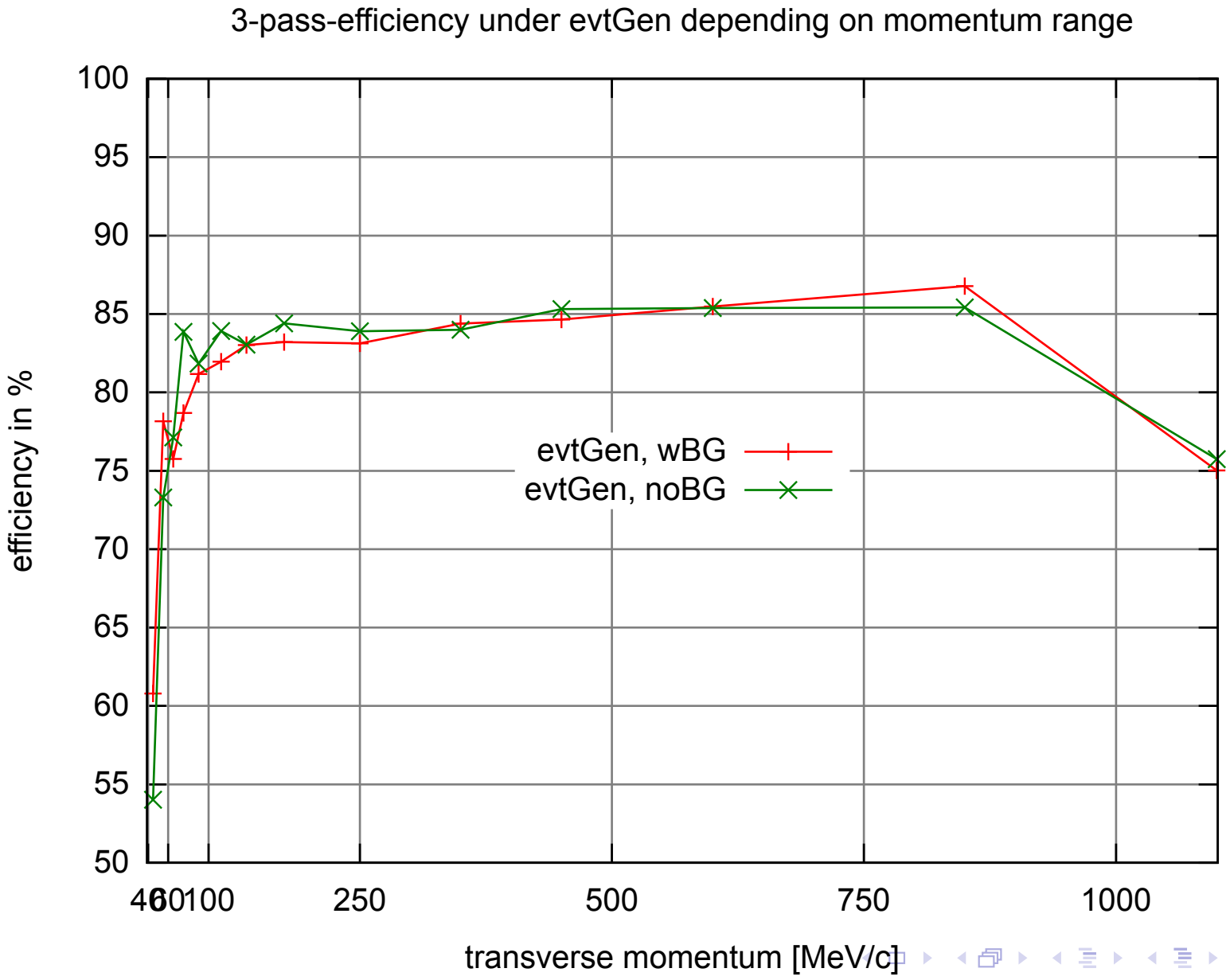


basic concept of cells, without sectorization restricted to sensor-wise combinations



extended concept using virtual segments attached to the IP and sectorMaps for segments in overlapping parts

efficiencies with evtGen



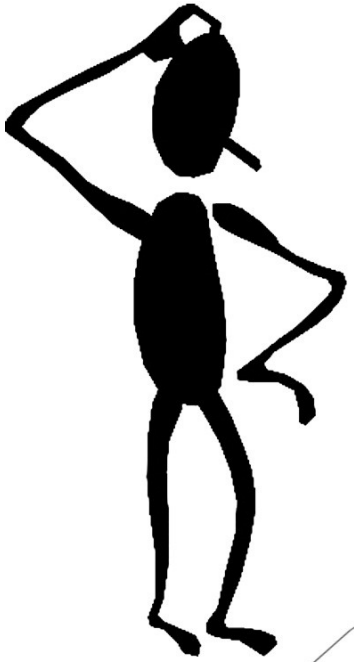
Conclusions

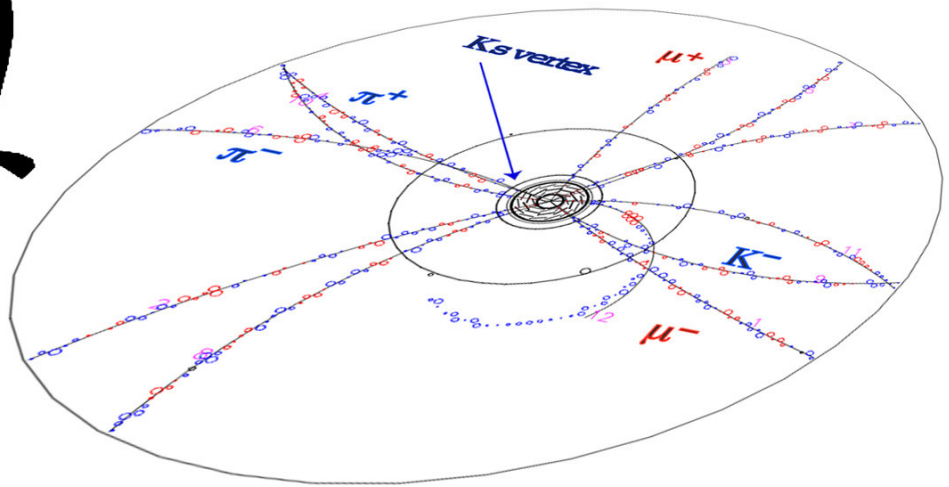
- CA is versatile and compatible with practically any geometry, not restricted to silicon detectors (initially developed for DCs, implementations for TPCs also known)
- Highly parallelizable since rule checks are independent from choices of neighbouring cells and updates are simultaneous
- Using lookup-tables the CA can be made more robust versus missing hits

Outlook

- ILD - project completed and fully working in the ILD-Framework
- Belle2 - further optimizations regarding curling tracks and performance are planned

that's all, folks!





Any suggestions, ideas or requests?

Jakob.Lettenbichler@oeaw.ac.at