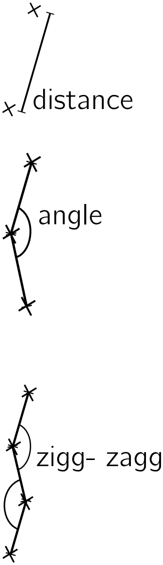
Current state and performance of the VXDTF

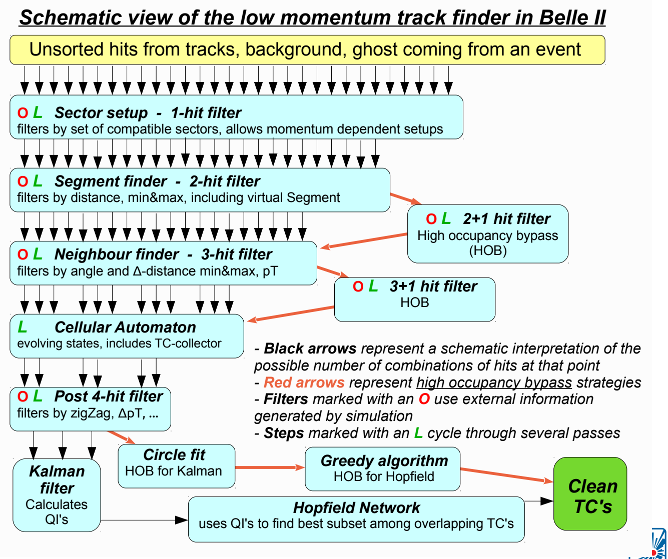
J. Lettenbichler, R. Frühwirth

Institute of High Energy Physics
Austrian Academy of Sciences

December 12, 2013

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 (HOB)

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

O L 3+1 hit filter
HOB

L Cellular Automaton
evolving states, includes TC-collector

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

Kalman filter
Calculates QI's

Circle fit
HOB for Kalman

Greedy algorithm
HOB for Hopfield

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

Clean TC's

- ***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*

For a better understanding...

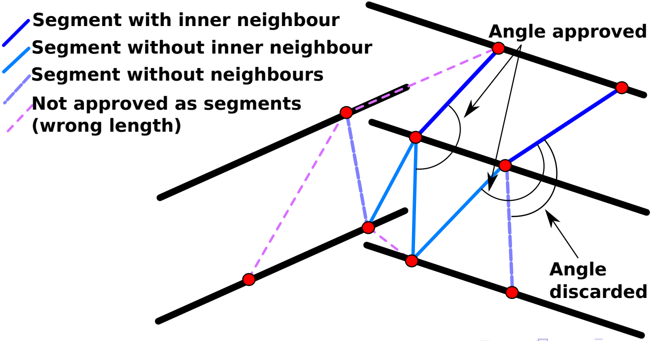
Vocabulary

- Sectors are sub-units of sensors with information about compatible pairs of sectors
- Compatible inner sectors are called ‘friends‘
- Cells of the Cellular Automaton are track segmens (short segments)
- Compatible inner segments are called ‘neighbours‘
- SecMap: lookup-table containing sector-friend-combinations and their individual filter-values
- Efficiency: tracking-efficiency compared to the MCTF (purity ¿ 70%)

Filters

Motivation using filters:

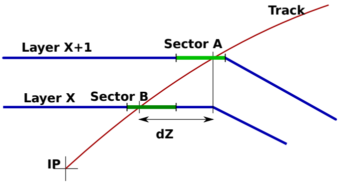
- Single hits are combined to segments which form TC's when connected → combinatorial problem
- Gradually filtering reduces combinatorics with increasing complexity
- Filter by cutoffs (2-hit: hit-distance, 3-hit: angle of linked segments)

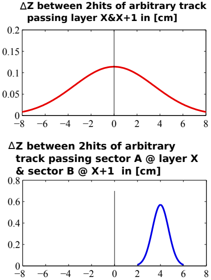

Segment with inner neighbour
Segment without inner neighbour
Segment without neighbours
Not approved as segments (wrong length)
Angle approved
Angle discarded

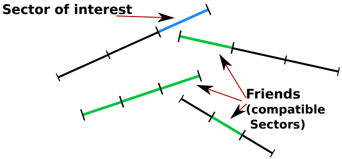
Sectors

Motivation using sectors:

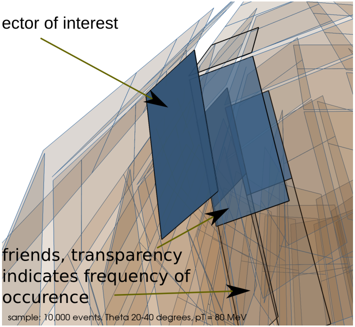
- Windmill structure and slanted sensors forbid simple layer-wise cutoffs → at least sensor-wise cutoffs
- Better: subdividing sensors in sectors and storing friend-lists
- → Allows customized cutoffs for filters to reduce combinatorics
- → Allows multipass optimizing for different momenta and curling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



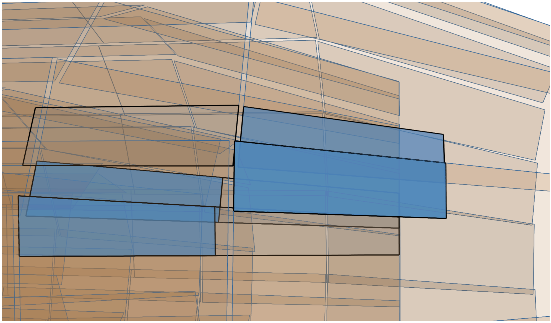




sectors and friends



sample: 10,000 events, Theta 20-40 degrees, pT = 80 MeV

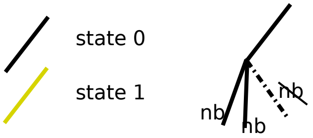


Concept: only hits are combined which lie in friend sectors (others don't get even checked → faster). Each sector carries independent information for each possible combination of friends

Side-effect: filters hits before entering the TF itself (reduces combinatorial work for segFinder)

Adapting CA principle for track finding

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



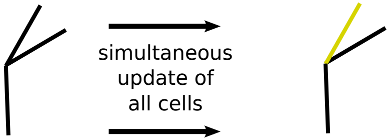
state 0

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 **Result**

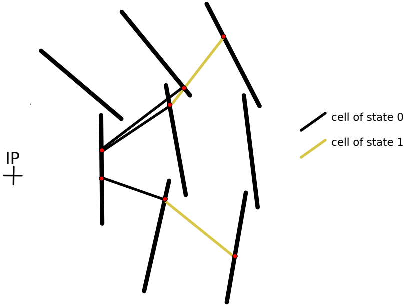


simultaneous update of all cells

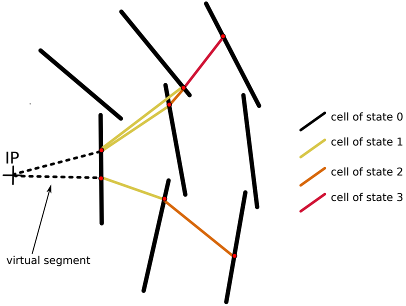
repeat steps until no cell evolves anymore

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

Adapting CA principle for 3-4 layers, virtual segment and sectors



Basic concept of cells



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

Creating TCs, QI-calculation and final TC-selection

- New TCs start with a seed (cells with high states), grows inwards by attaching cells with decreasing value of state
- A TC-Filter applies simple rules like zigg-zagg or Δp_T
- The quality index (QI → probability that current TC is a real track) of a TC can be calculated by the following algorithms
    - Track length
    - Kalman filter (genfit2)
    - Circle Fitter
- These QIs are used to define a non-overlapping subset of TCs by using one of the following algorithms
    - Neuronal network of Hopfield type (highest reconstruction rate)
    - Simple greedy algorithms (faster, worse quality)

Outlook (taken from F2F-slides of June '13)

Still some bugs which are difficult to trace down

- Not always working filter calculation (extensive studies needed, low priority)
- Θ dependency of efficiency (especially around 90°) issue unresolved (extensive studies needed)
- ~~Huge memory consumption due to XML-related bug (currently waiting for bugfix/switch to SQL) - workaround needed? would take some time~~
- ~~Hopfield produces strange behavior in rare occasions which are difficult to reproduce(e.g. accepting overlapping TCs or killing whole set of TCs)~~
- ~~Pass-merging-process is a bottle-neck in time consumption, will be resolved by new version including real hit removal steps (no hit removal so far which produced many ghost tracks, especially in low p_T)~~

Outlook

- Bugfixing
- ~~Testbeam preparation next months~~
- Starting implementation of vectorized combinatorial Kalman filter which is official main goal of PhD-Thesis (will be implemented next to current VXDTF)
- Another detailed studies of BG-events, when there are more BG-types available

(taken from SVDPXD-slides of October '13)

To do:

- Having sectormaps for covering any setup to be tested during the testbeam (who is officially responsible for implementing testbeam geometry-cases to basf2?)
- ~~Residuals for ROI-finding-seeds and track fitting have to be rechecked, current values strange~~
- Thorough testing of genfit2 compatibility (including new KF implementation using reference tracks), → to be done after genfit2 release

Testbeam preparation - What has been done so far

Already done:

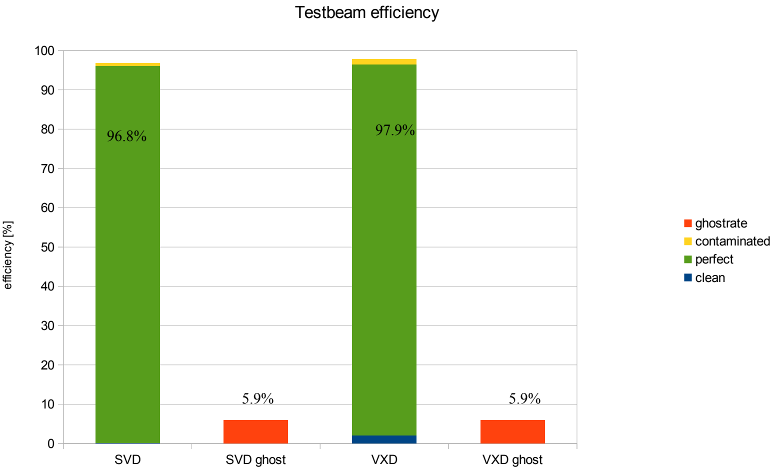
- Adapting VXDTF for the testbeam case, 2 & 6 GeV
- Calculating new lookup tables for the testbeam setup (uploaded some to the testbeam package including example file)
- Fixing old bugs in the TF (Hopfield, memory leaks, inefficient procedures)
- Speeding up the TF (SVD: 150 µs/tbEvent, VXD: 175 µs/tbEvent)
- Implementing baseline TF intended for low occupancy events including support for SVD-hits with missing 2nd cluster
- Workaround for the problematic XML-parsing, reduced memory consumption by a factor of 4 at least (for testbeam - whole basf2 framework needs now 160 MB)

More progress

Other issues I am working on:

- Complete replacement of old way to create sectorMaps - 95% done (feature complete, but bugfix necessary)
- Restructuring the internals of the VXDTF to be able to connect the CKF - ∼70%
- VXDTF-validation - 50% (got the internal protocolling and my local root script, but unable to get it into validation Plot-thing)
- CKF design and implementation - 10% (thanks to genFit2 is a baseline-approach easy to implement, but current priority at testbeam)
- Constant refurbishing of the code - XX% (never-ending story)

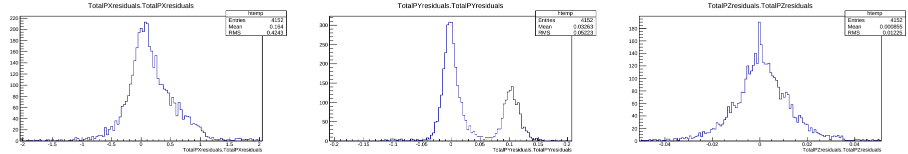
Efficiency for testbeam setup



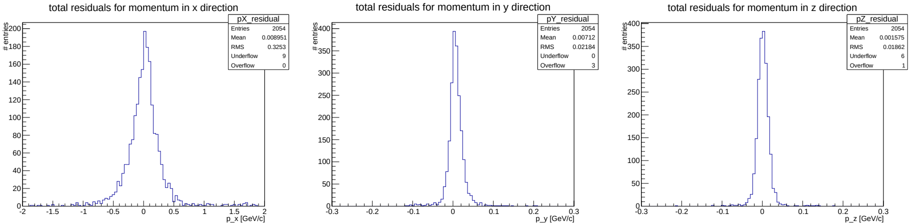
setup: r7086, testbeam/vxd/example/VXDTFdemoTB.py, 5000 events, 2GeV positrons, 0.976T

TB 2GeV residuals for p_x, p_y & p_z

r7086(PXDSVD Hamburg Oct 2013):



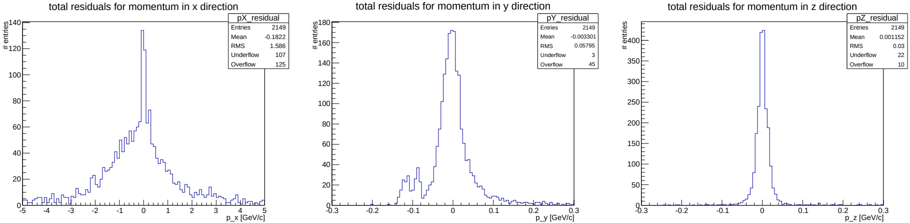
r7172 (fix for residuals-bug):



all TCs which were accepted as good ones (more than 70% of their hits were of the same track), units [GeV/c]

TB 6GeV residuals for p_x, p_y & p_z

not uploaded yet: 6 GeV



all TCs which were accepted as good ones (more than 70% of their hits were of the same track), units [GeV/c]

Outlook

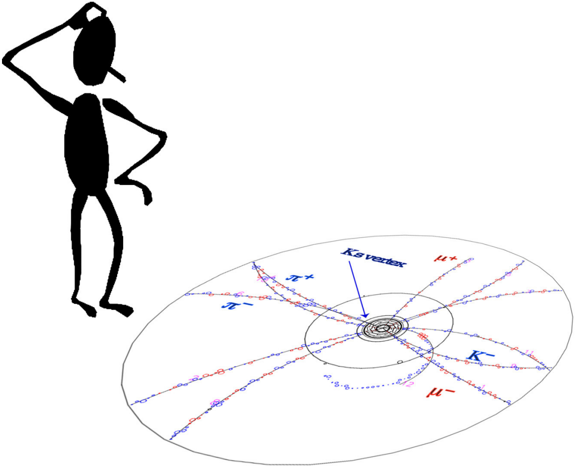
Questions/Info from my side:

- Who is the basf2-geometry guy I am working with?
- How do I get updated sectormaps onto the HLT? (SVN, Flash drive, Voodo?))
- I will be at DESY at January 13th-24th
- What kind of on-line monitoring tools do you wish to be implemented?

To do:

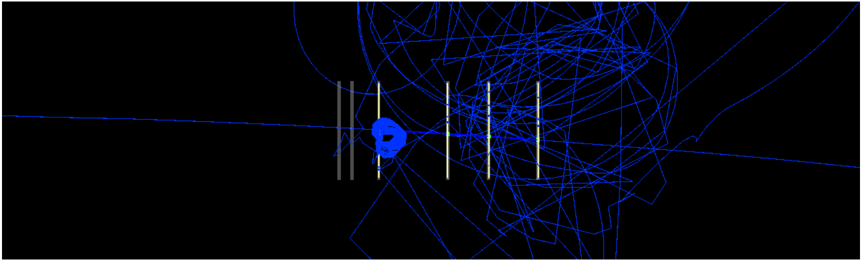
- Testbeam
- VXDTF - increase quality, efficiency and readability, more testing procedures
- CKF (baseline, improved + vectorized), strongly relying on genfit2 (baseline using full geometry, improved using detplanes based on sectors, analytical track propagation (constant B)
- DAF

that's all, folks!



Any suggestions, ideas or requests?
Jakob.Lettenbichler@oeaw.ac.at

Why not 100%? - this one was lost



up to 268 hit combinations per sensor...

Well, this one was found