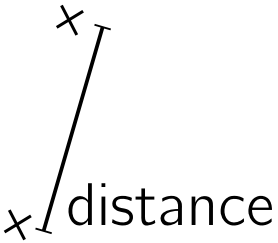
F2F Update for low momentum trackfinder June 2013

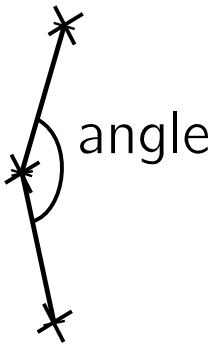
J. Lettenbichler, R. Frühwirth

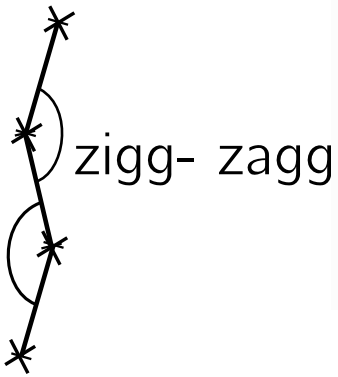
Institute of High Energy Physics
Austrian Academy of Sciences

June 21, 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*

Done since last F2F meeting

- Added second (useful) TC-set-cleaner (greedy algorithm: fast, but worse quality of final result compared to Hopfield NN)
- Overhaul of internal structure (extensive use of classes instead of functions, process still ongoing)
- Finishing current Reimplementation of Pass-merging-process (old version was extremely slow, but new version is still a bottle-neck)
- Further speed optimizations (especially useful for high occupancy cases)
- Bugfixes
- Maintaining compatibility to recent changes in FW
- Quasi stable version for several months now (but no users any more since Moritz left, nothing VXDTF-related ever heard of Tabuk, new hope: Pisa, genFit2-tests)
- Time consumption acceptable for any current case possible (SVD, VXD, with or w/o BG)

Current issues (selection)

Still some bugs which are difficult to trace down

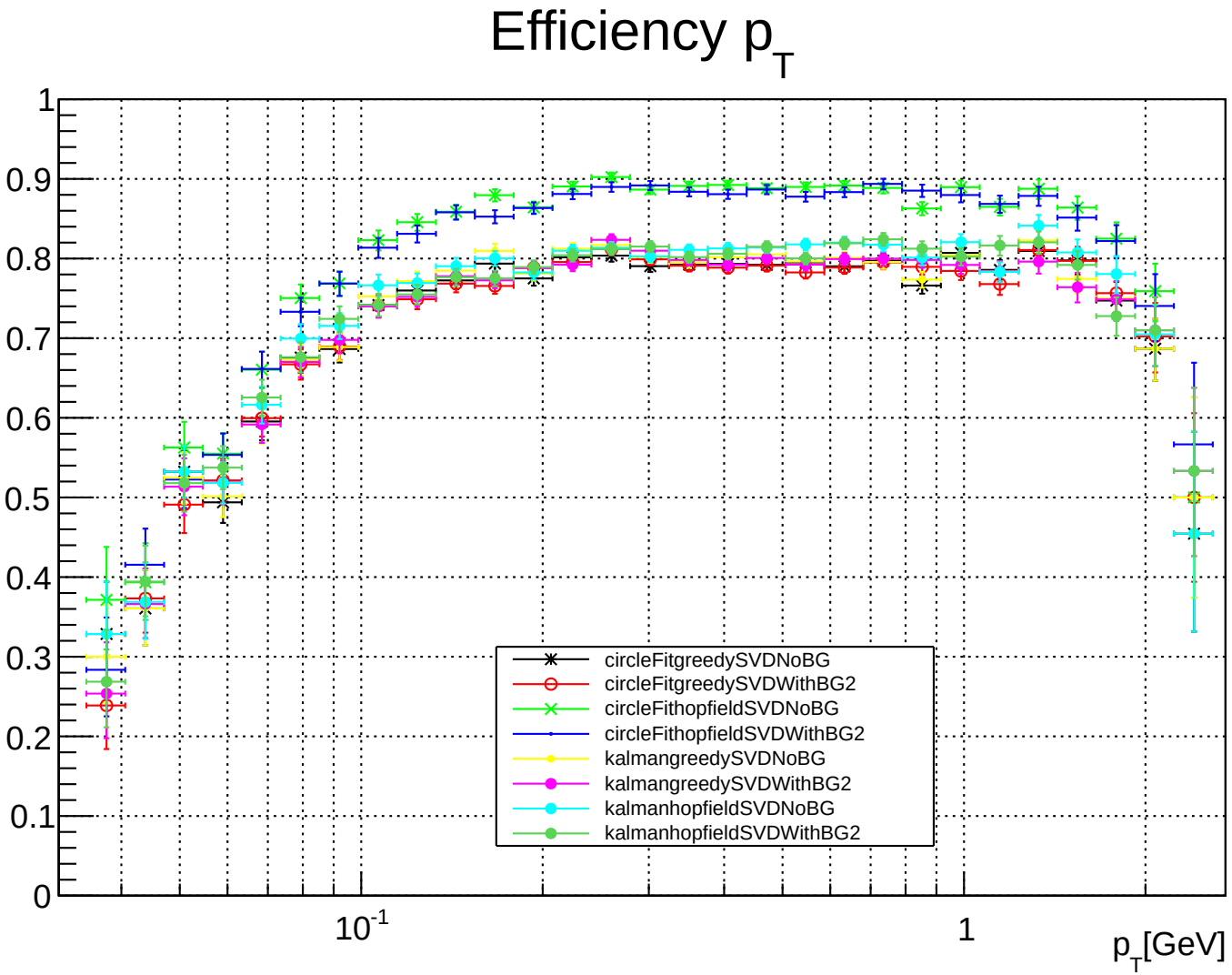
- Not always perfectly working filter calculation (extensive studies needed, currently no time for that, low priority)
- Theta dependency of efficiency (especially around 90°) issue unresolved (extensive studies needed too,
- huge memory consumption due to XML-related bug (currently waiting for bugfix/switch 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)
- Even new version of 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 momentum)

Efficiency issue

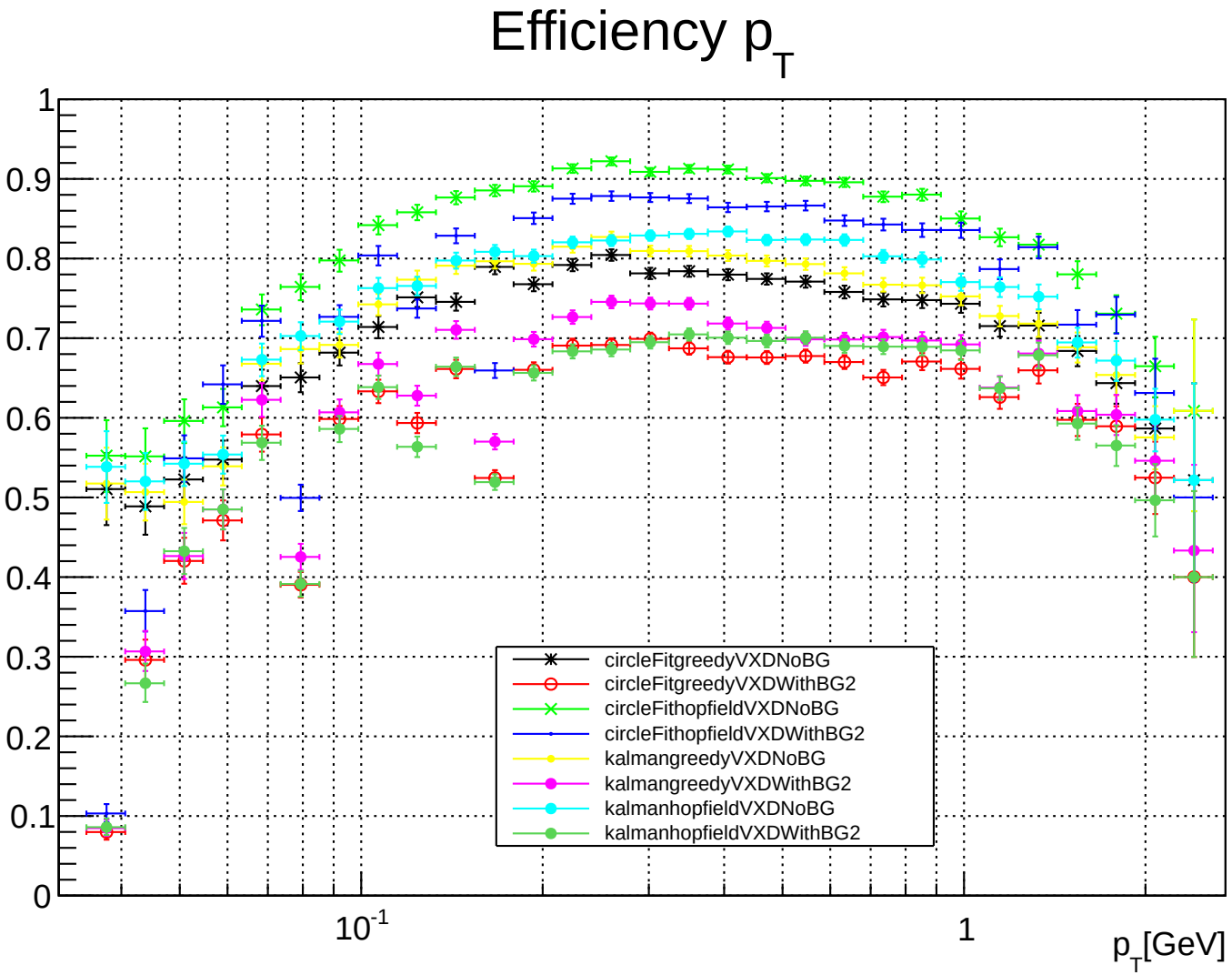
Why not 100%

- Undiscovered bugs
- Deviations in results due to TrueHit → Cluster differences (difficult to solve due to nontrivial connection between TrueHits and Clusters, should reimplement procedure using new relations which are much easier to handle)
- Slanted parts-bug still existing
- Particle not always activating both sides of SVD-sensors (about $\sim 5\%$ of all tracks have got at least one hit with missing 2D info)

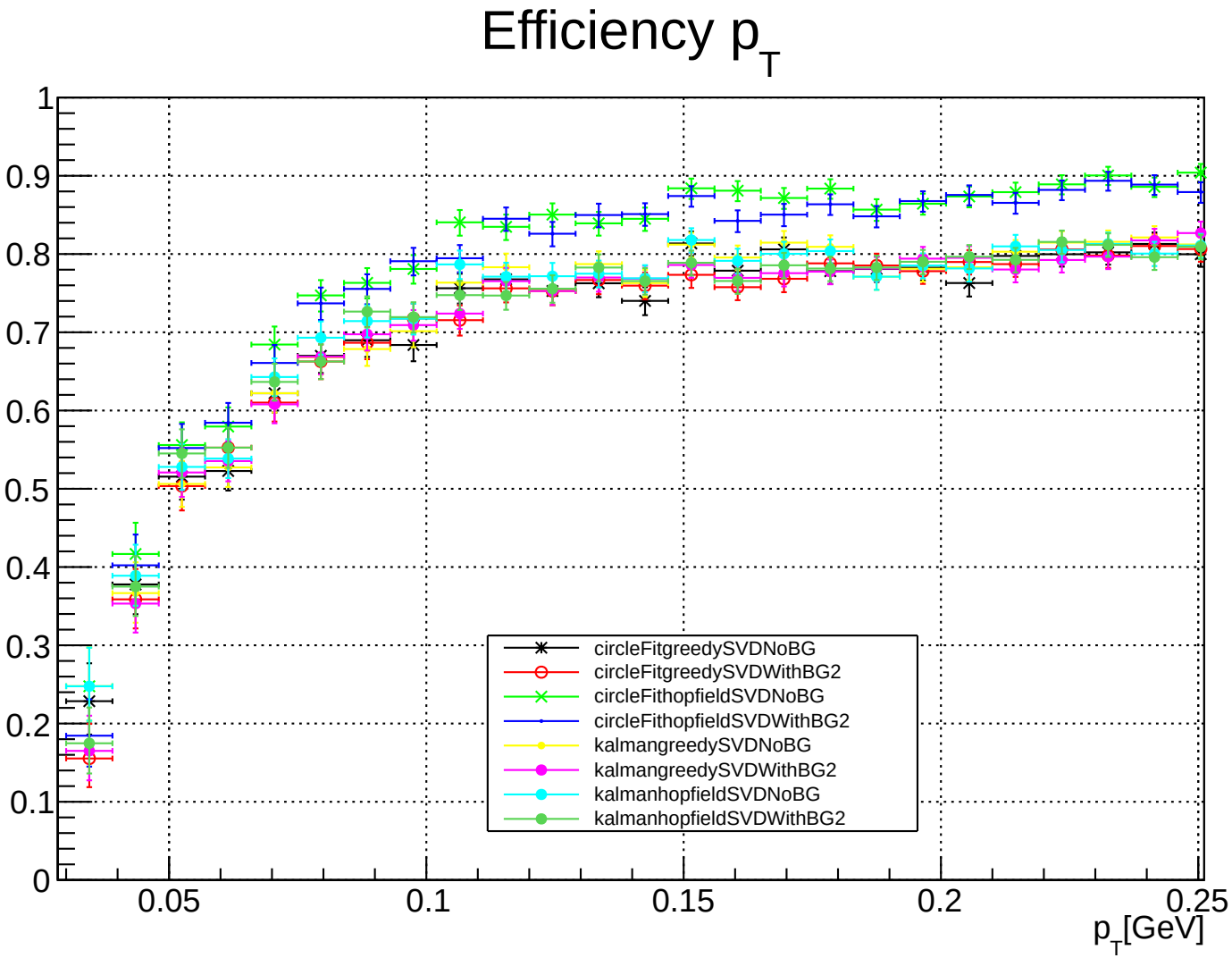
efficiencies with evtGen - jun13 SVD, w+w/o BG



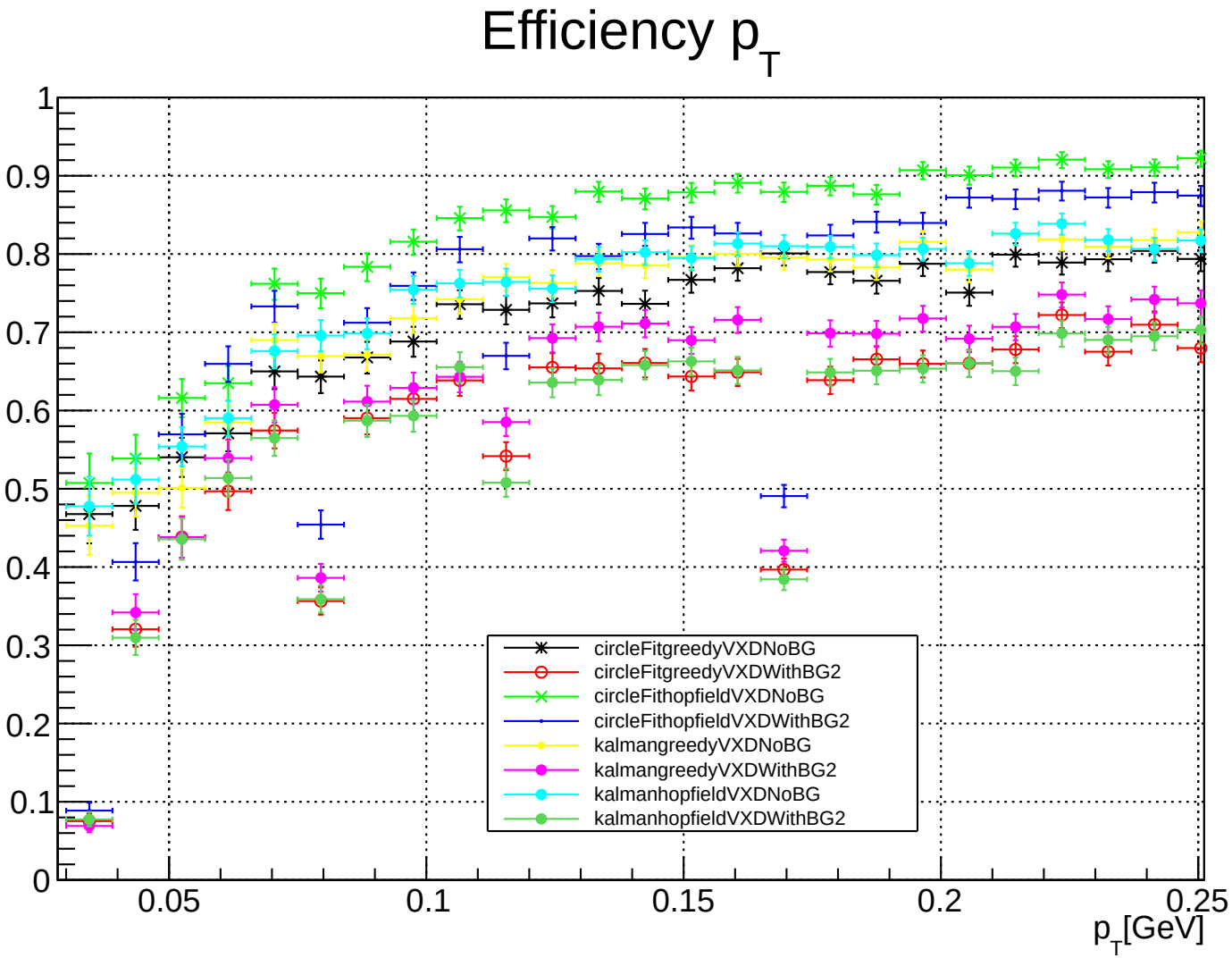
efficiencies with evtGen - jun13 VXD, w+w/o BG



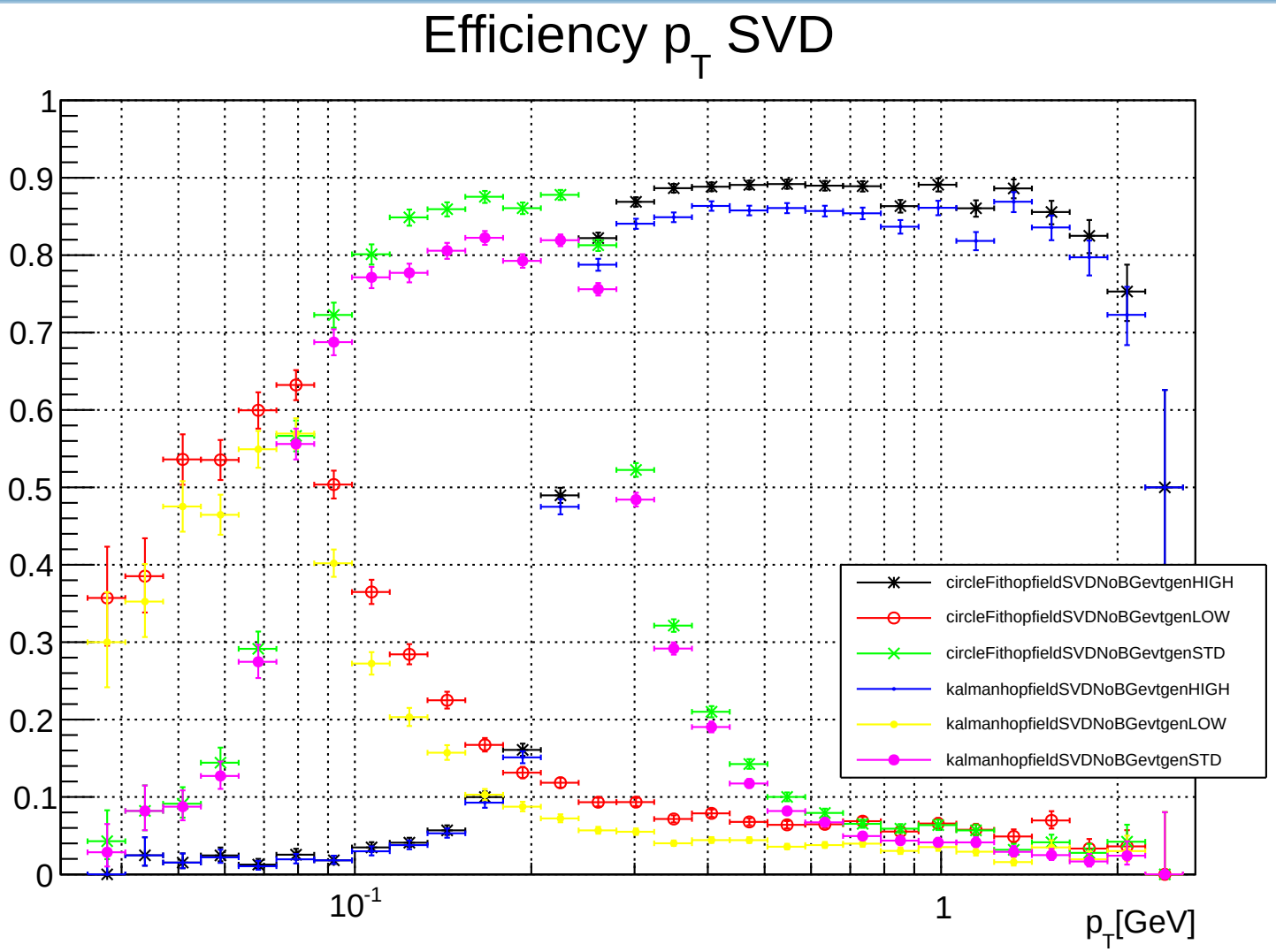
efficiencies with evtGen - jun13 SVD lowPt, w+w/o BG



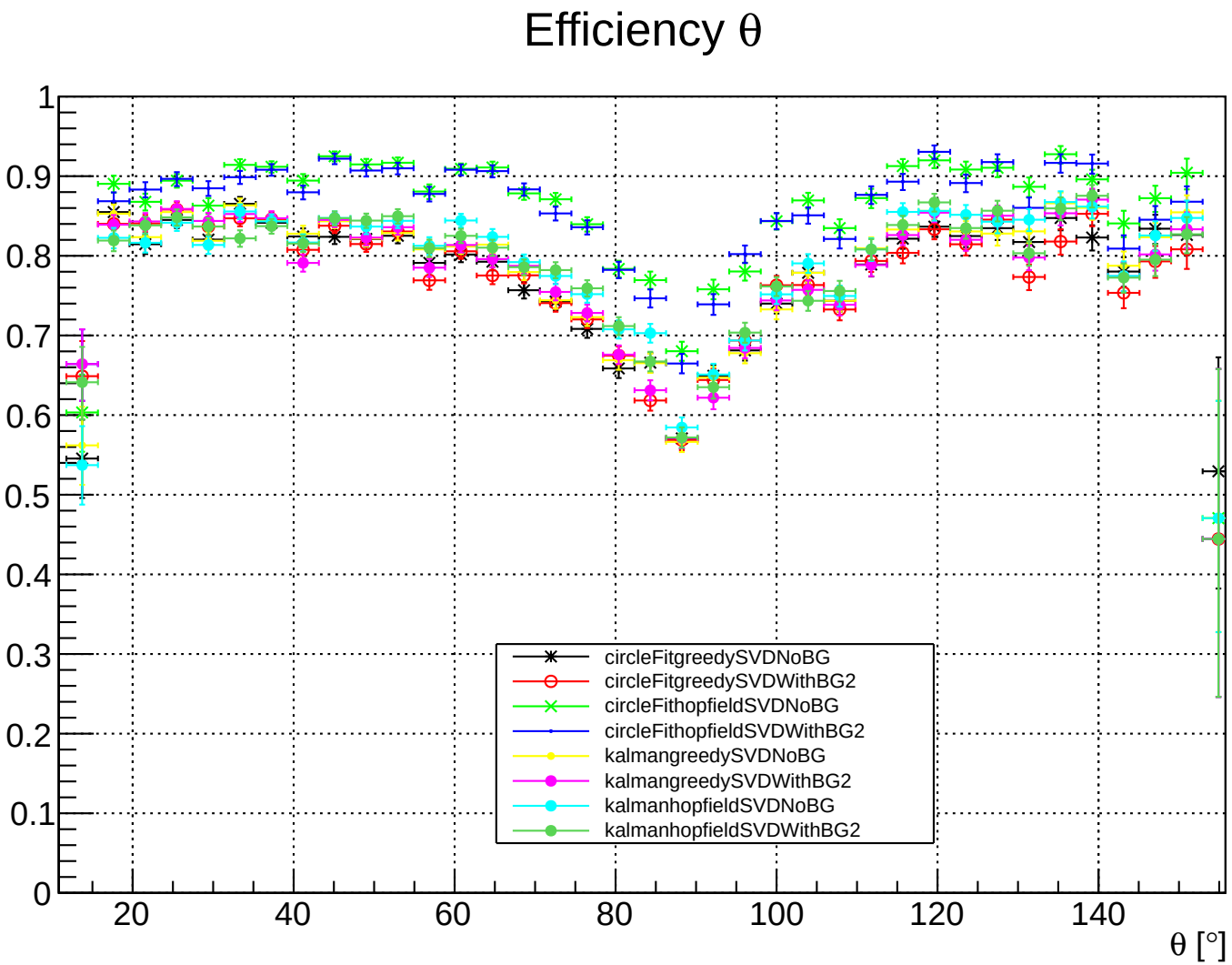
efficiencies with evtGen - jun13 VXD lowPt, w+w/o BG



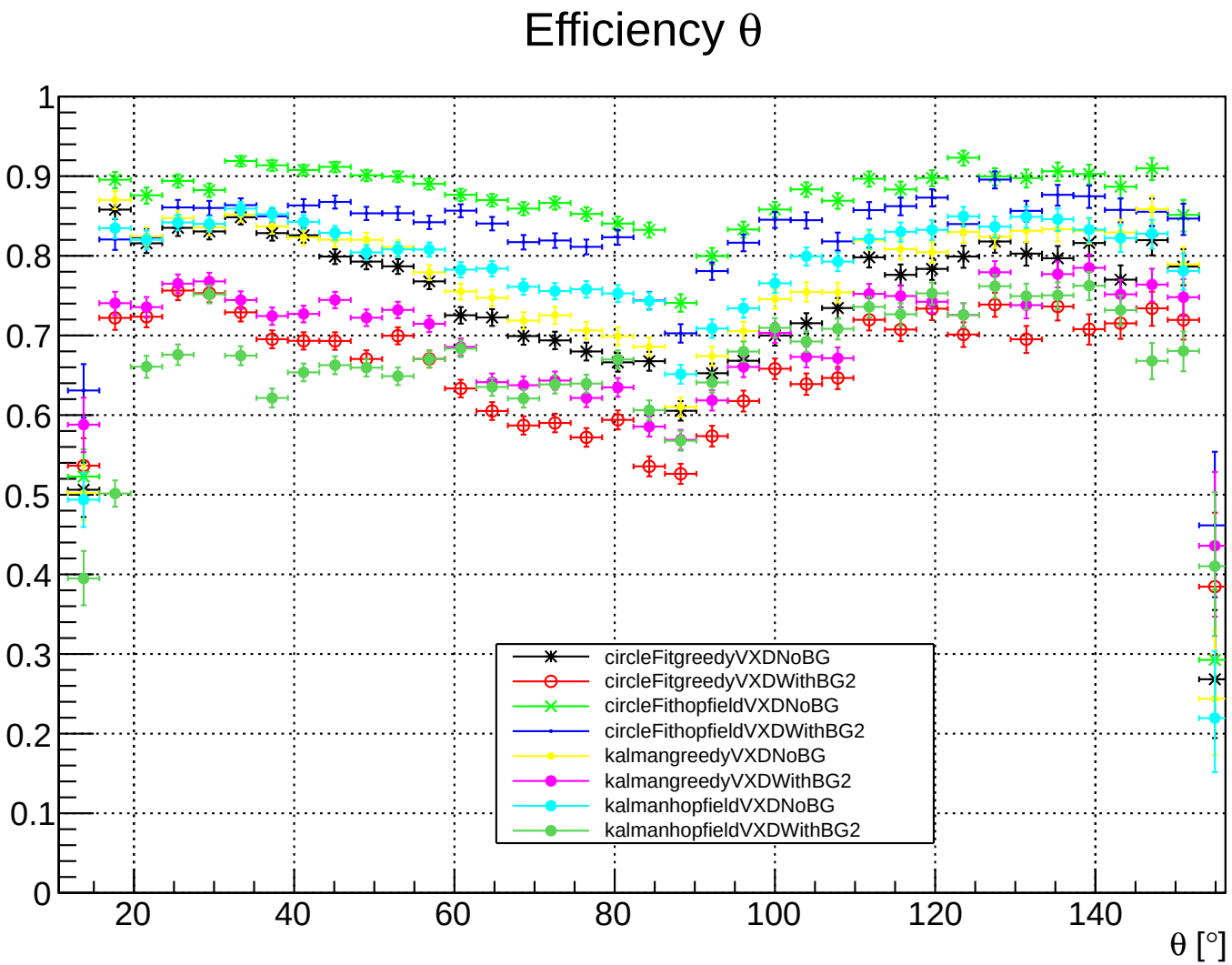
efficiencies with evtGen - jun13 VXD singlePass, w/o BG



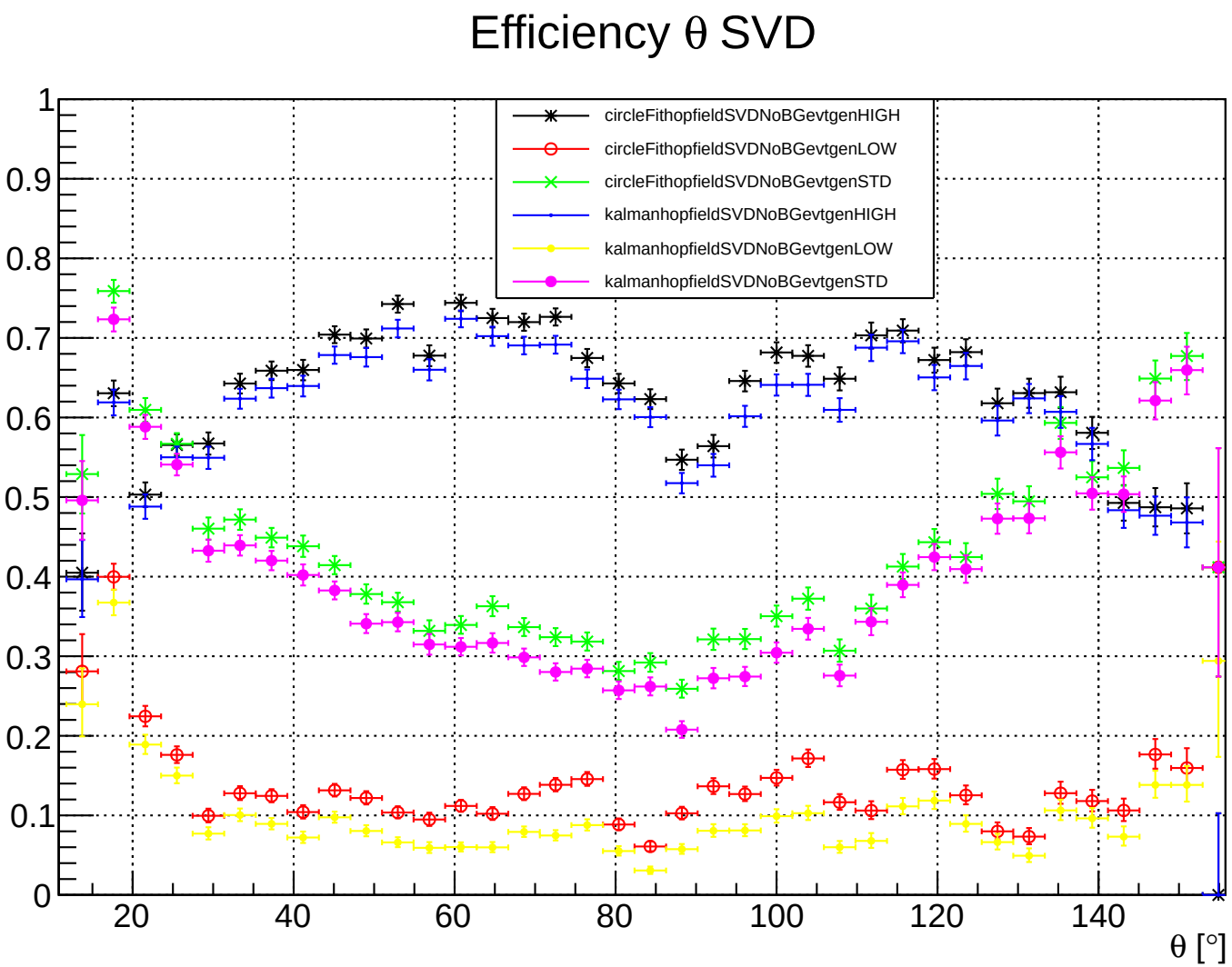
evtGen - dependency of θ svd, w+w/o BG



evtGen - dependency of θ vxd, w+w/o BG



evtGen - dependency of θ single pass, w/o BG



Time consumption

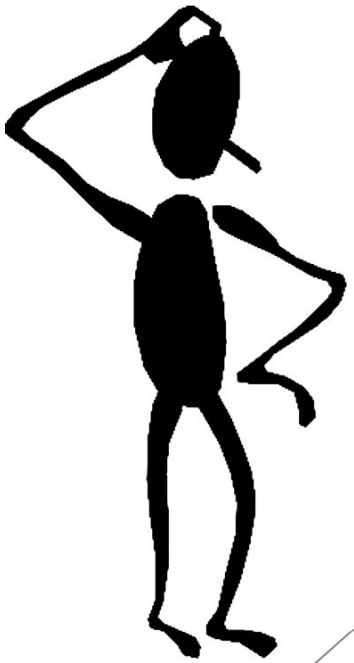
SVD-noBG:
CF+GR 77.5%/14.0% 44ms
CF+NN 86.6%/22.7% 47ms (opt-mode: 6ms)
KF+GR 78.2%/13.3% 101ms
KF+NN 79.3%/11.3% 103ms
SVD-withBG:
CF+GR 77.4%/13.9% 43ms
CF+NN 86.0%/22.5% 47ms
KF+GR 77.9%/13.3% 99ms
KF+NN 78.9%/11.0% 102ms
VXD-noBG:
CF+GR 75.3%/9.8% 34ms
CF+NN 87.3%/25.3% 37ms
KF+GR 77.3%/8.2% 112ms
KF+NN 79.4%/7.1% 115ms
VXD-withBG:
CF+GR 48.2%/26.2% 133ms
CF+NN 61.0%/53.3% 140ms
KF+GR 50.9%/23.1% 240ms
KF+NN 48.8%/17.1% 245ms

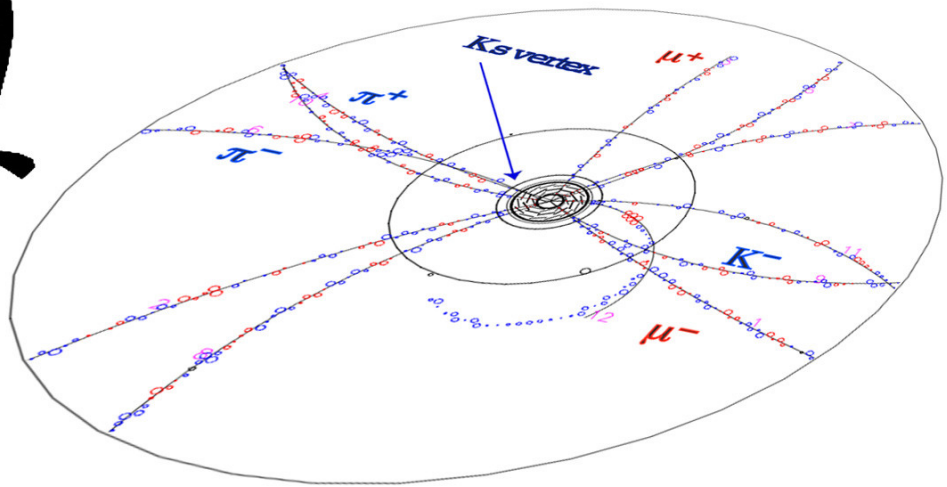
Outlook

Outlook

- Bugfixing
- Another overhaul of pass-merging-process including hit removal this time
- Testbeam preparation next months
- Starting implementation of vectorized combinatorial Kalman filter which is official main goal of PhD-Thesis (will be implemented in current VXDTF as another option)
- Another detailed studies of BG-events, when there are more BG-types available

that's all, folks!





Any suggestions, ideas or requests?

Jakob.Lettenbichler@oeaw.ac.at