Status of VXDTF-related modules

J. Lettenbichler, R. Frühwirth

Institute of High Energy Physics
Austrian Academy of Sciences

September 29, 2014

What happened since last time?

Been there, done that:

- Finished working on the beam test issues
- Finished supervising Bachelor student Thomas Fabian
- Continued supervising Project student Stefan Ferstl
- Learning for some exams
- Study about the behavior of the VXDTF for cases with and without background
- Started redesign of the VXDTF

Beam Test at Desy Hamburg - January 2014

Track finding was possible

- Online support for 4-layer SVD-Tracking
  - First contact with real sensors and realistic alignment
  - Fast enough for the mini HLT ($< 200\,\mu s$/event)
  - Worked with and without magnetic field
  - Issues with L3, very bad efficiency (alignment? unknown VXDTF-error?)
- For off-line: Support for SVD, PXD and TelClusters, please use:
  - Use VXDTF for normal tracking and TB too (example files for more details)
  - Use BaseLineTF (testbeam package) for minimalistic track finding for TB cases

Progress with supervised students

- Thomas Fabian:
  - Complete check for filters used for the VXDTF
  - Now (partially) automated report generation for the analysis of filters for given run/setup
  - Extremely slow (3 hours for 400 events - calculations done in C++, Python and R), but very helpful
  - most relevant new insight 1 - all filters work as expected
  - most relevant new insight 2 - filters are *not* the cause for the drop of efficiency at $\theta \approx 90^o$
- Stefan Ferstl:
  - Implemented an interface for internal VXDTF-Logging with output for analysis tools and display module
  - Work basically completed, currently waiting for write-up

- $p_T = 250\,MeV/c$, realistic Clusters and Background

Example for result of $60° \leq \theta \leq 70°$:

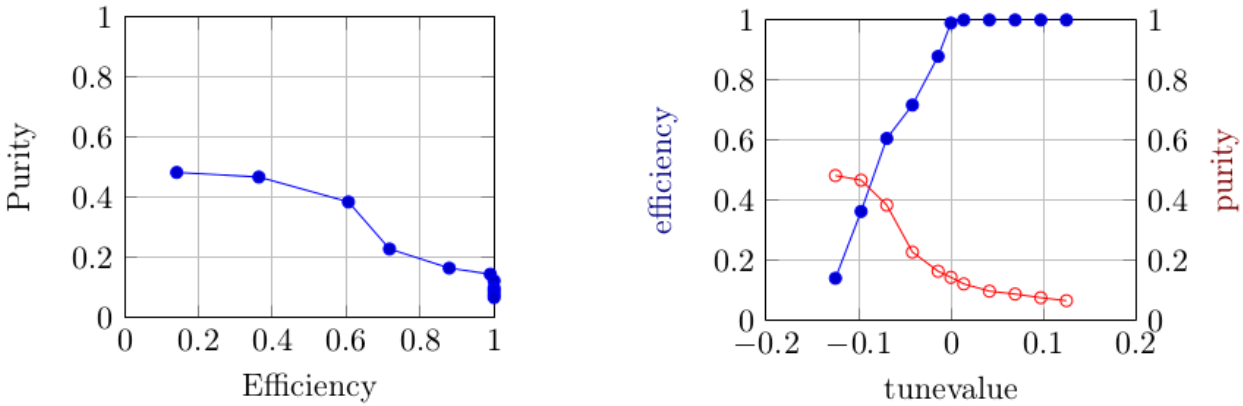


Figure 8: distance3D

Example for result of $85° \leq \theta \leq 95°$:

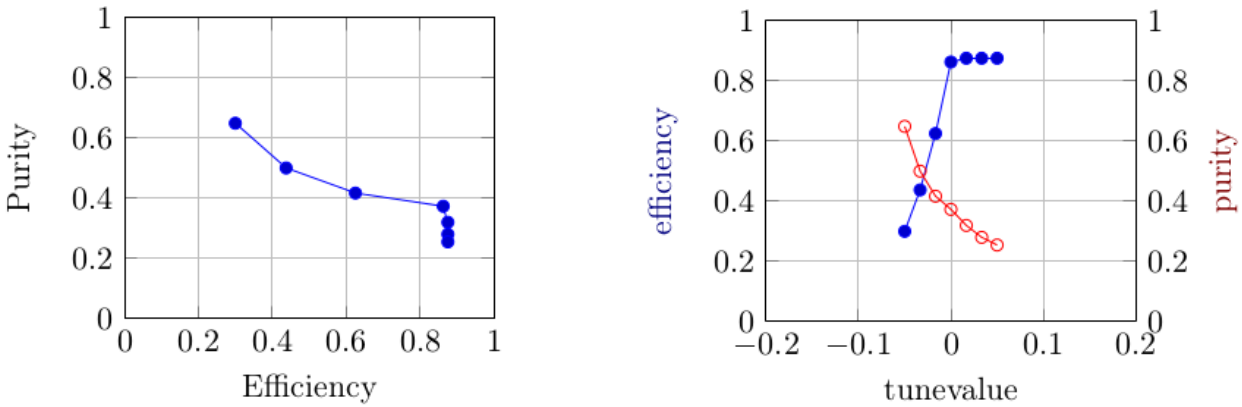
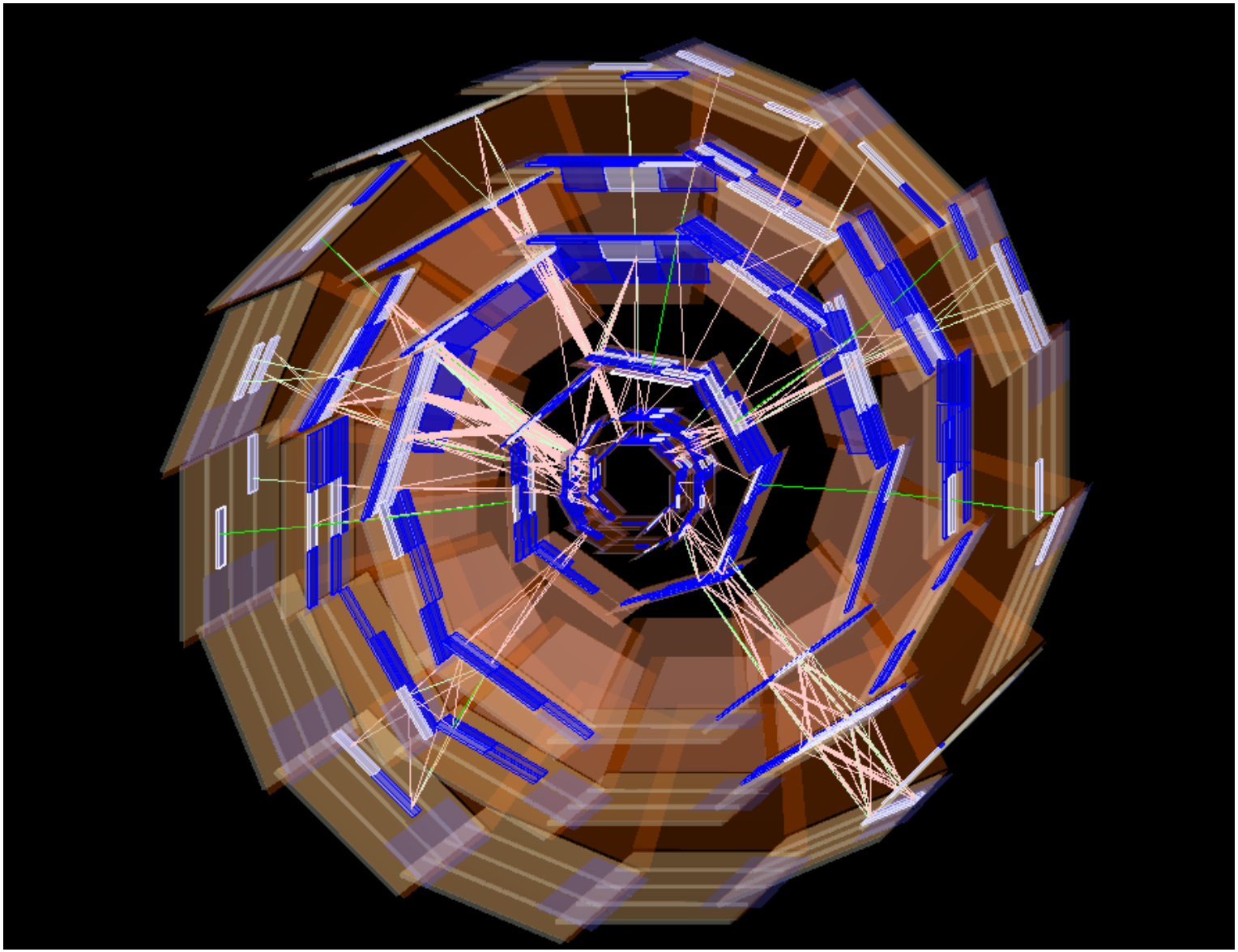
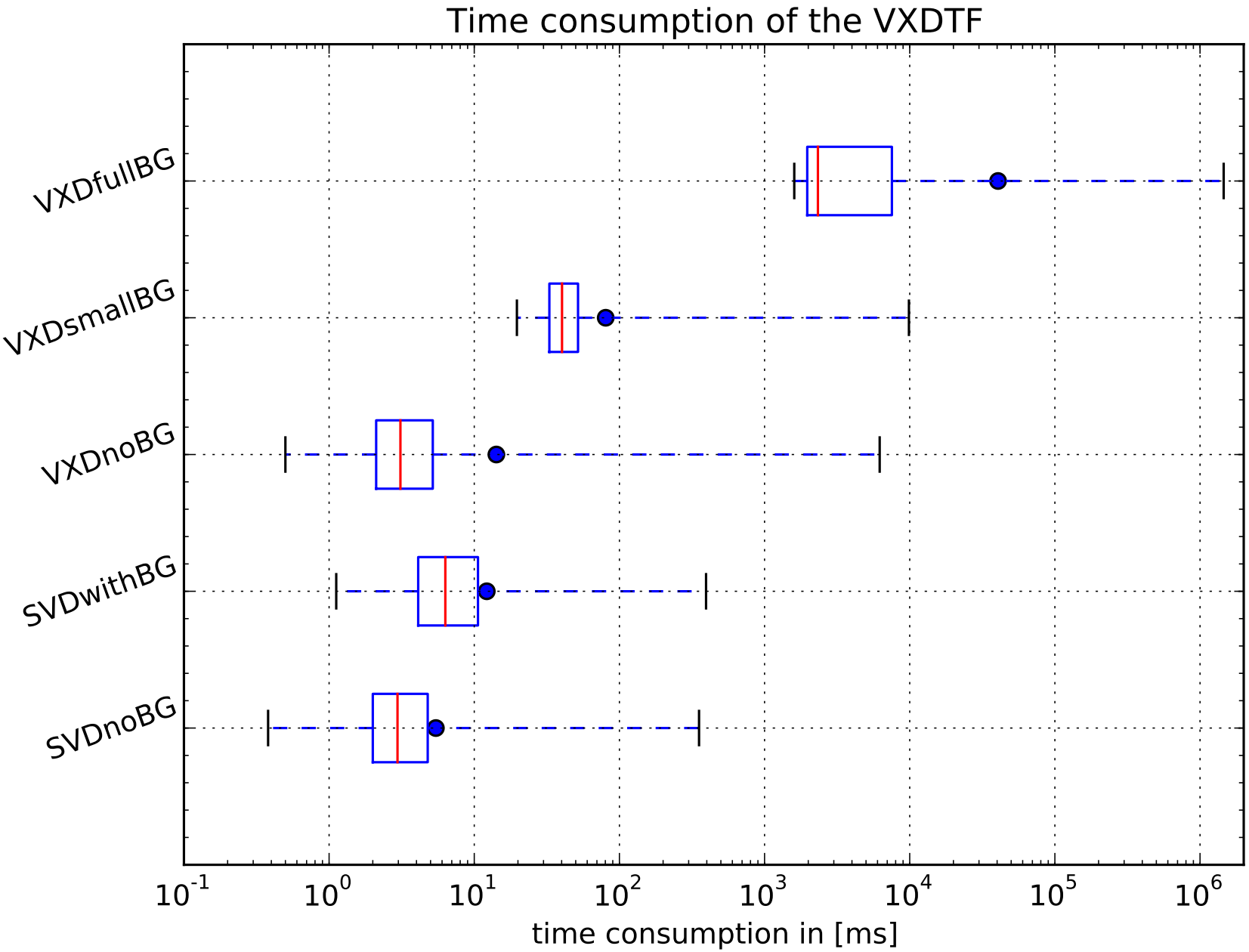


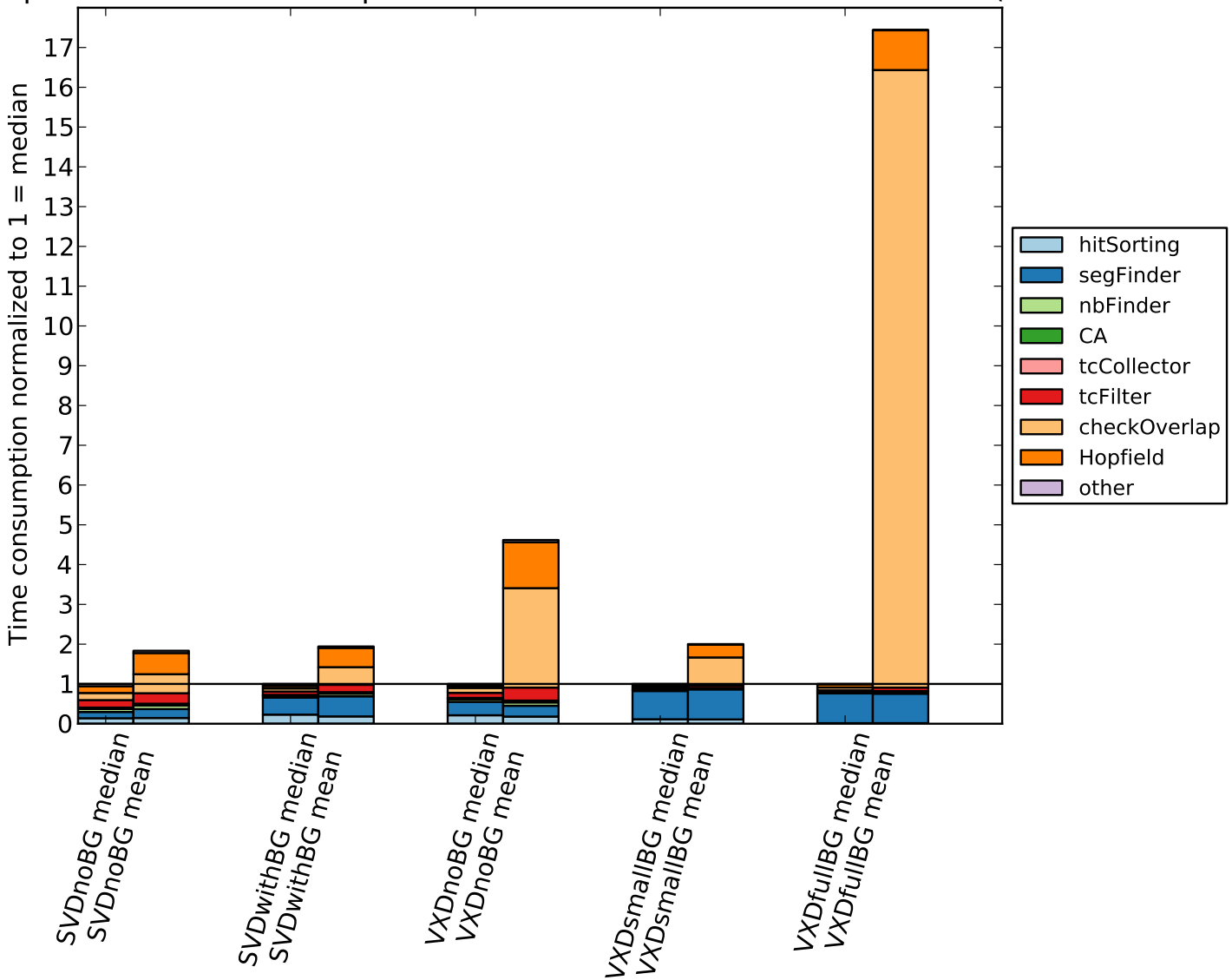
Figure 7: distance3D

Typical event w/o BG

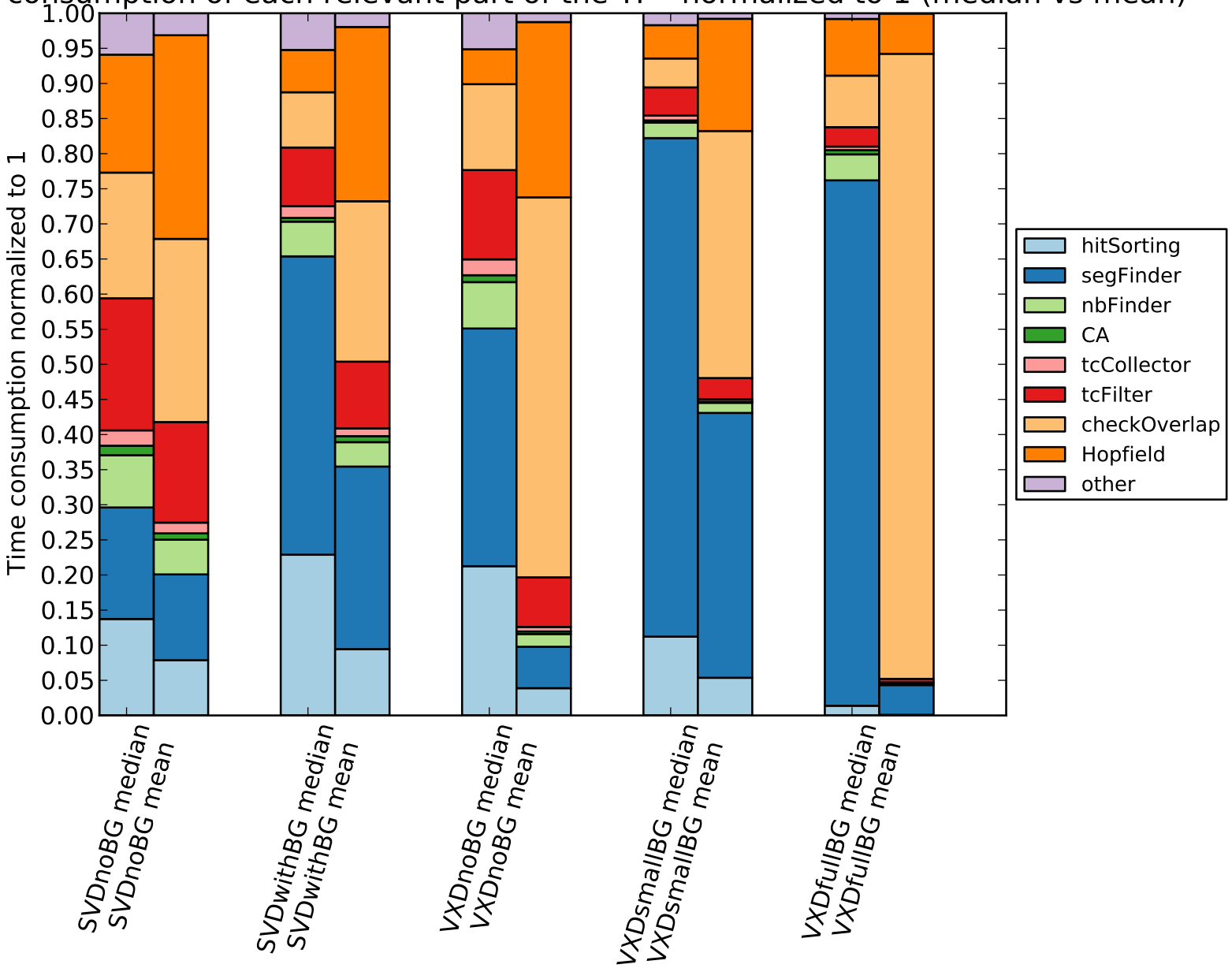



Time consumption of the VXDTF
VXDfullBG
VXDsmallBG
VXDnoBG
SVDwithBG
SVDnoBG
10^{-1}
10^{0}
10^{1}
10^{2}
10^{3}
10^{4}
10^{5}
10^{6}
time consumption in [ms]

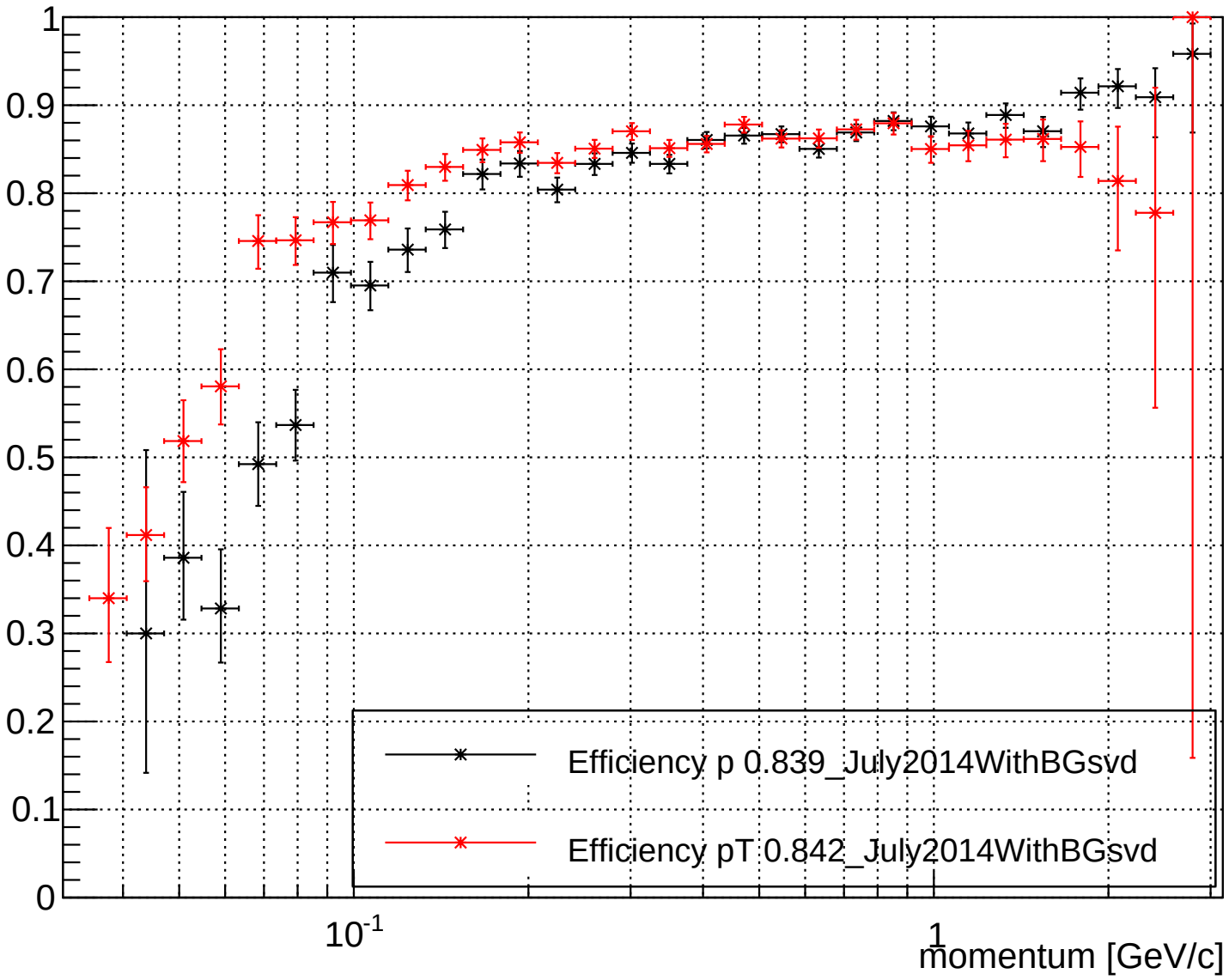
Time consumption of each relevant part of the TF - normalized to 1 = median (median vs mean)



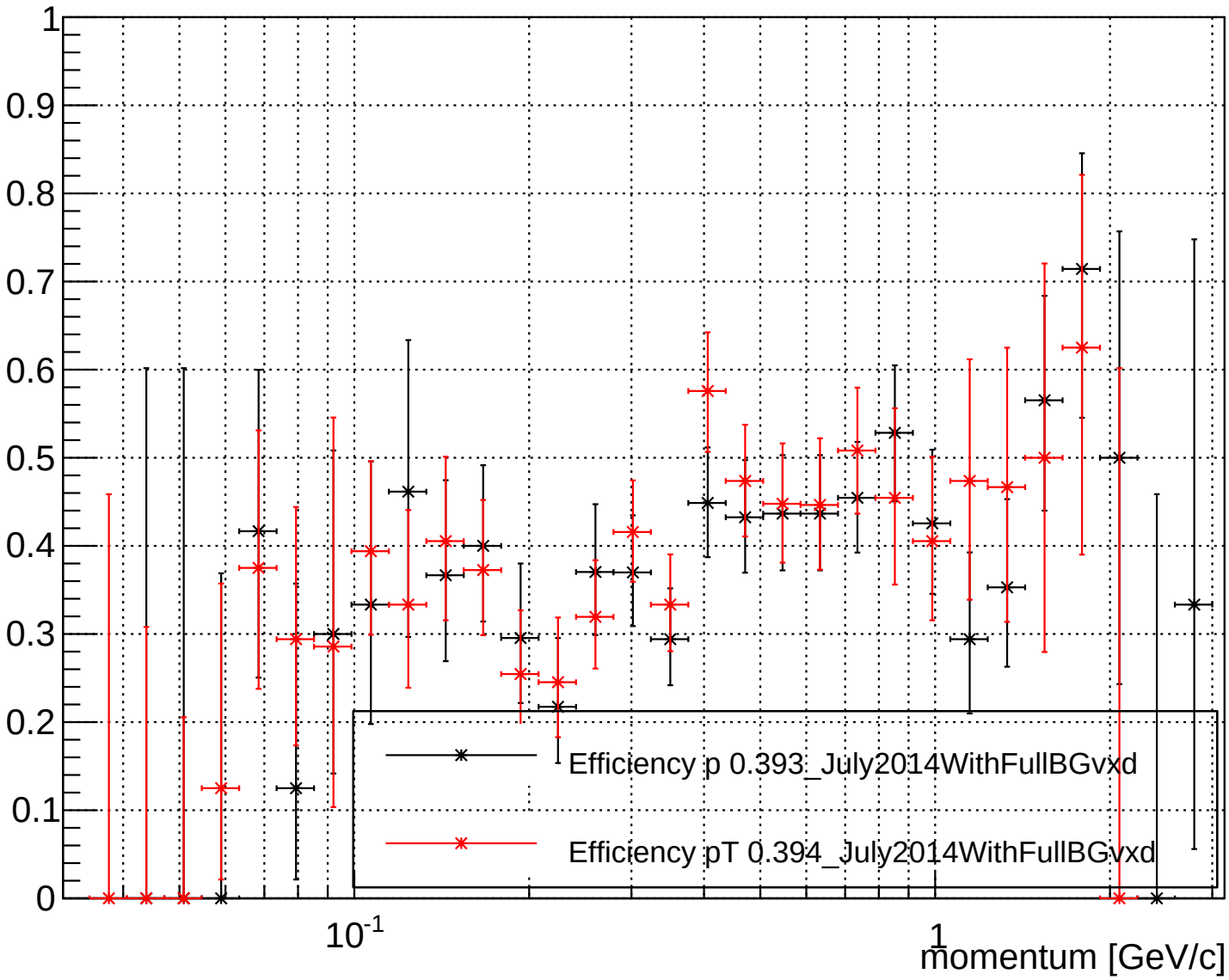
Time consumption of each relevant part of the TF - normalized to 1 (median vs mean)



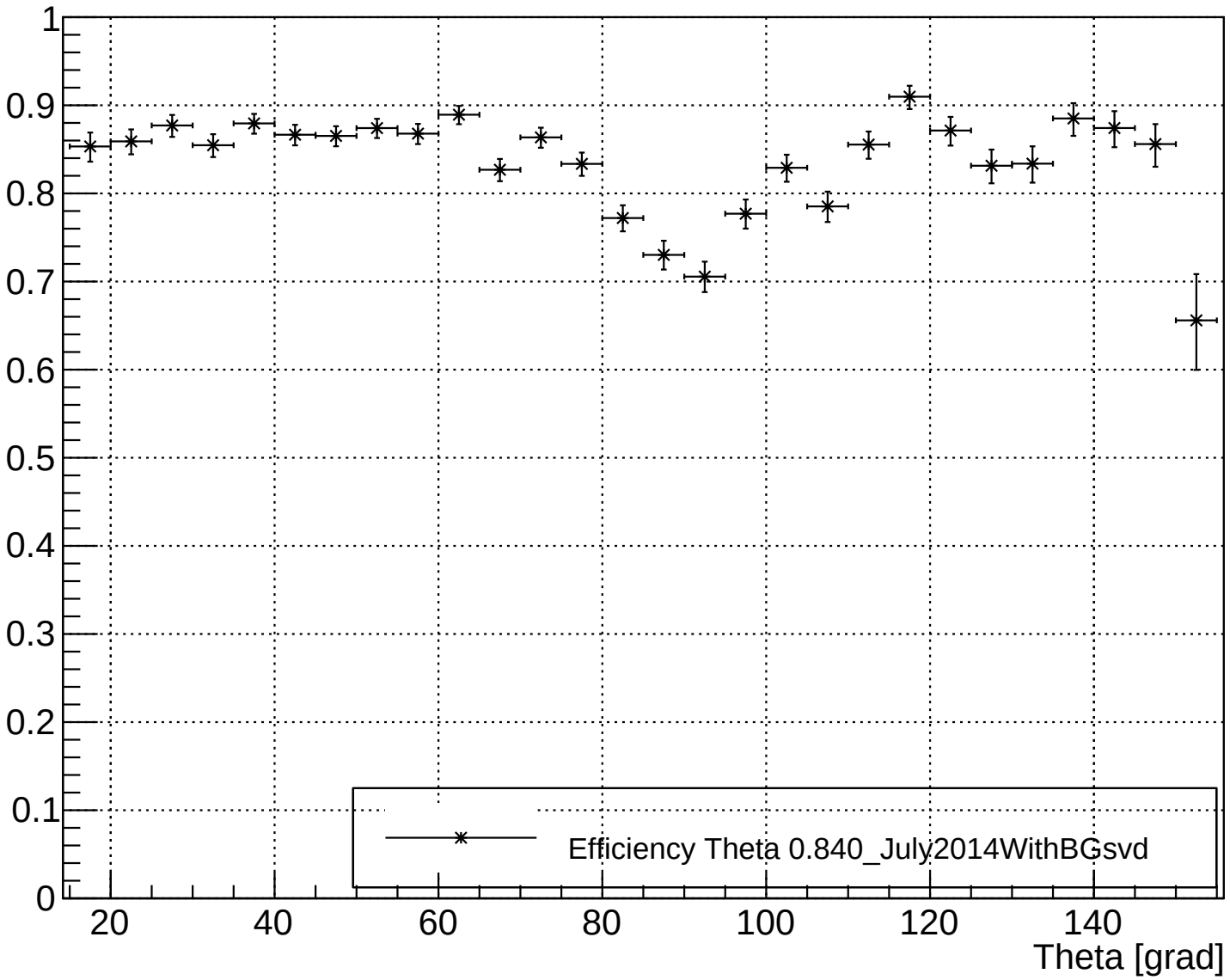
Efficiency vs momentum



Efficiency vs momentum



Efficiency vs theta



Redesign

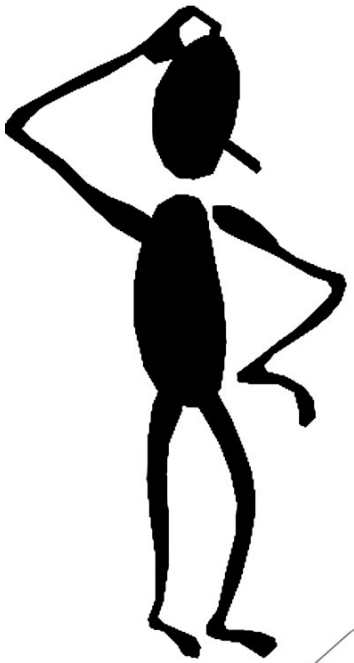
Just a few remarks

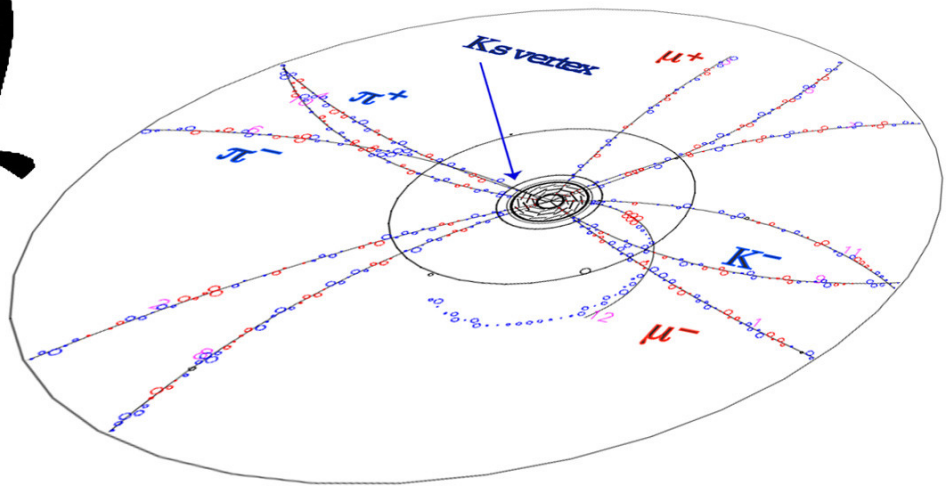
- 20500+5000 bare lines of VXDTF-related C++-code (counted without comments and empty lines)
- SpacePoints mostly done - storing metaData not finalized yet
- The VXDTFModule is now frozen in its interface, can safely be used
- Work is done in the TFRedesignModule which will change names after redesign is done
- No immediate performance improvements should be expected right after finishing redesign
- Still ongoing

Things you should keep in mind when using VXDTF

- Currently there is an issue causing low efficiency for the VXDTF in the PXD:
  - Seems to be independent from Clusterizer
  - Some tuning changes for setup did not help
  - Problem will be adressed after redesign (end of year)
- some severe bugs were removed from the VXDTF not so long ago, therefore please use current versions of the code
  - Monthly releases do still carry the old code (yet) - please refrain to use that code for studies
  - r12481 or newer contain the fix for efficiency (except the PXD-problems mentioned above)
  - r12875 or newer contain all fixes relevant so far - including most of nondeterministic behavior

that's all, folks!

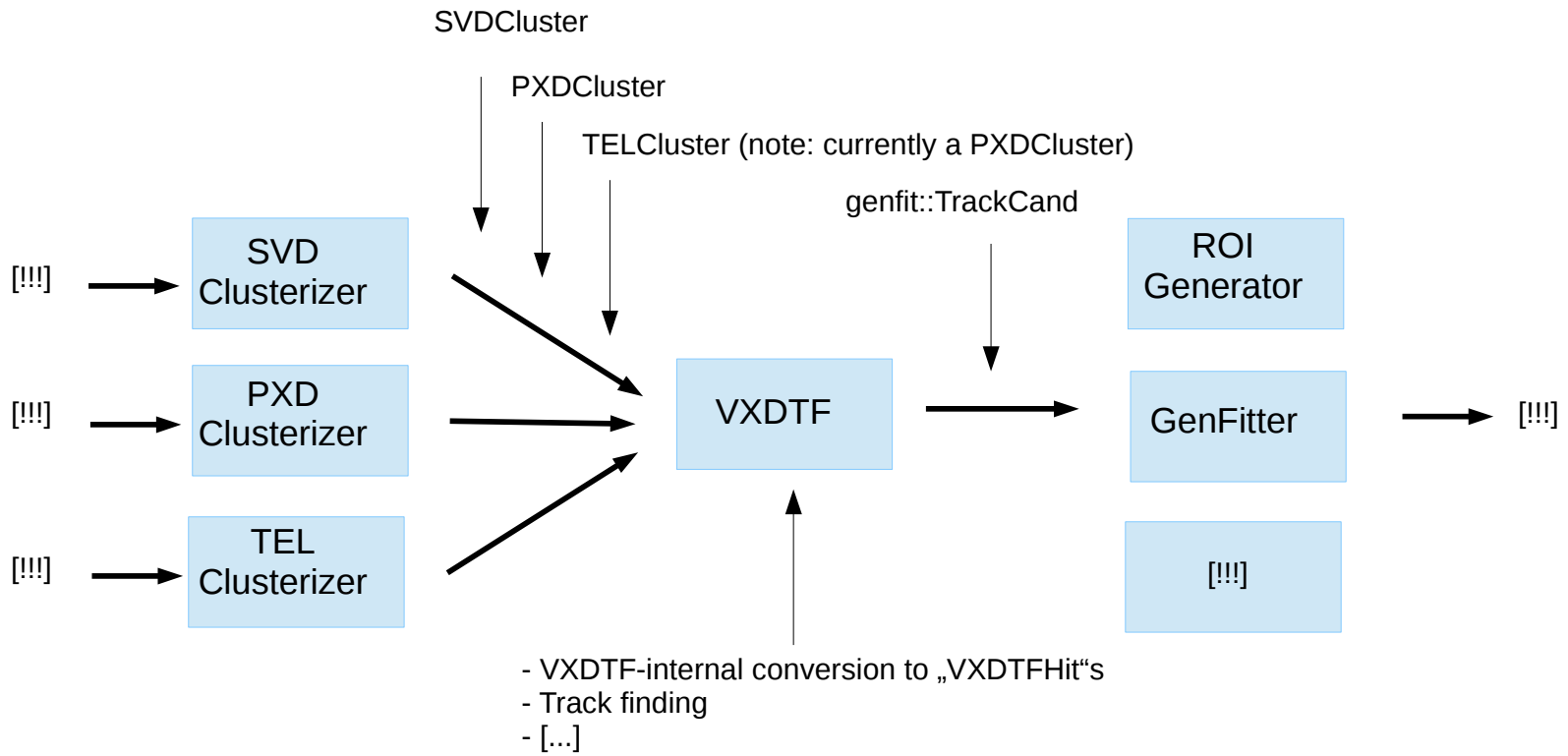




Any suggestions, ideas or requests?

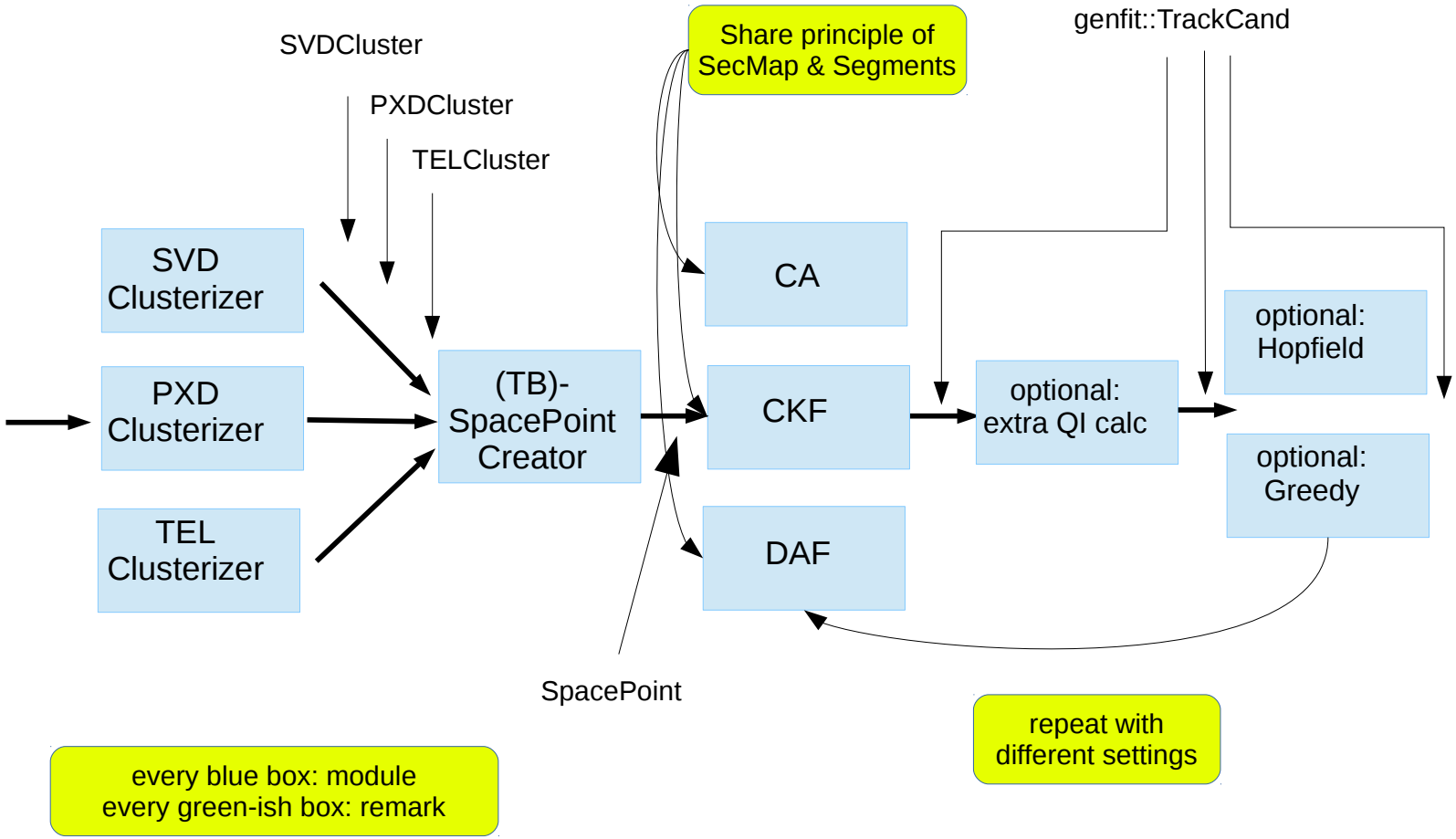
Jakob.Lettenbichler@oeaw.ac.at

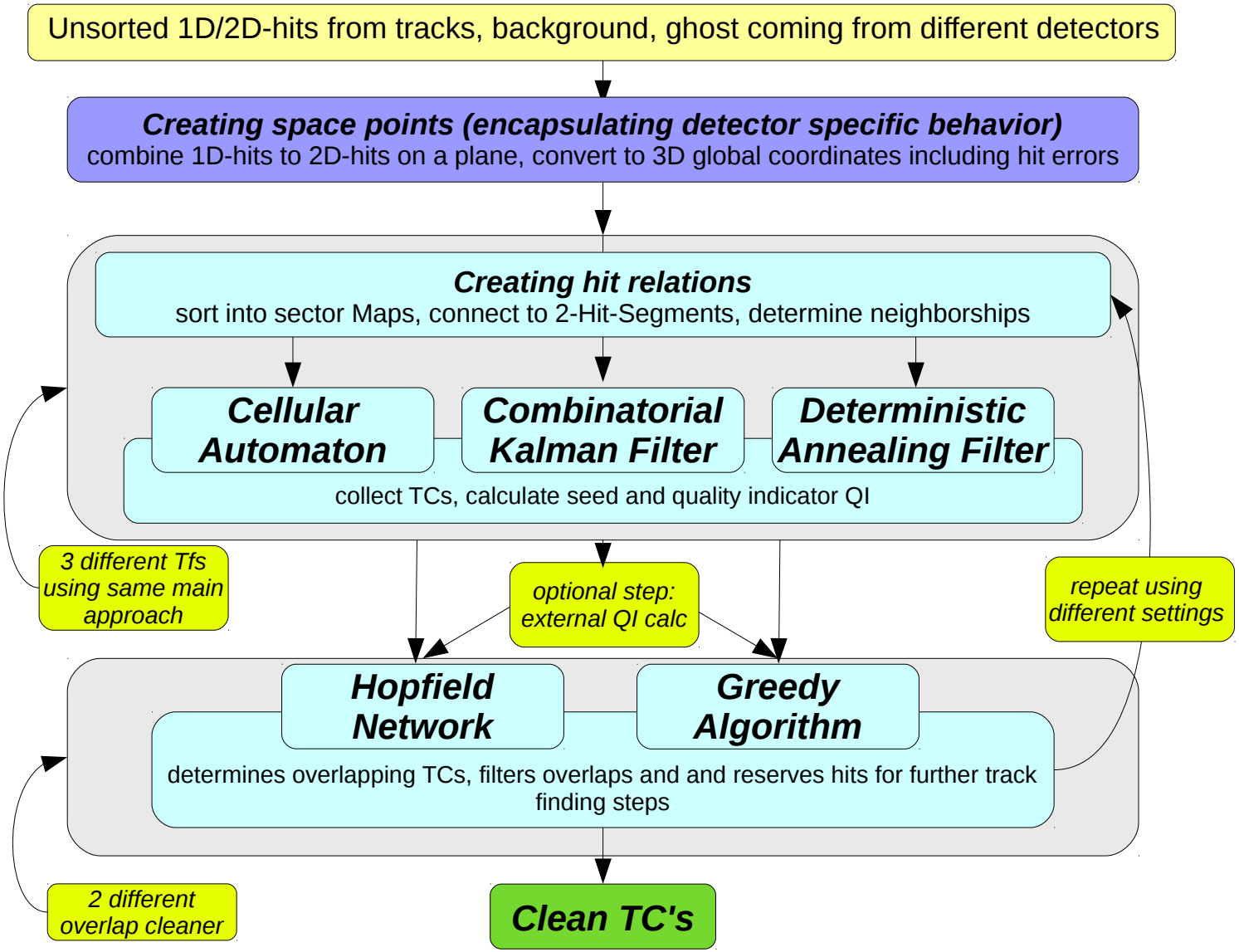
Current state of this part of the reconstruction chain



[!!!] : shortcut for „a lot of important stuff which is not part of this discussion“

Future state of the trackFinder VXD-approach (event-part)




Unsorted 1D/2D-hits from tracks, background, ghost coming from different detectors
Creating space points (encapsulating detector specific behavior)
combine 1D-hits to 2D-hits on a plane, convert to 3D global coordinates including hit errors
Creating hit relations
sort into sector Maps, connect to 2-Hit-Segments, determine neighborships
Cellular Automaton
Combinatorial Kalman Filter
Deterministic Annealing Filter
collect TCs, calculate seed and quality indicator QI
3 different Tfs using same main approach
optional step: external QI calc
repeat using different settings
Hopfield Network
Greedy Algorithm
determines overlapping TCs, filters overlaps and and reserves hits for further track finding steps
2 different overlap cleaner
Clean TC's