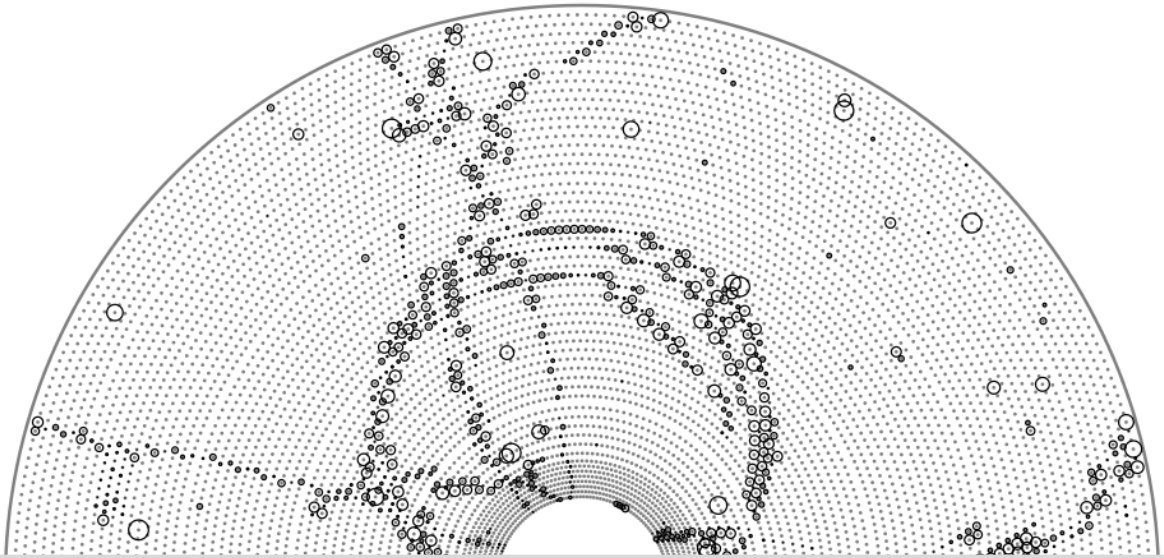
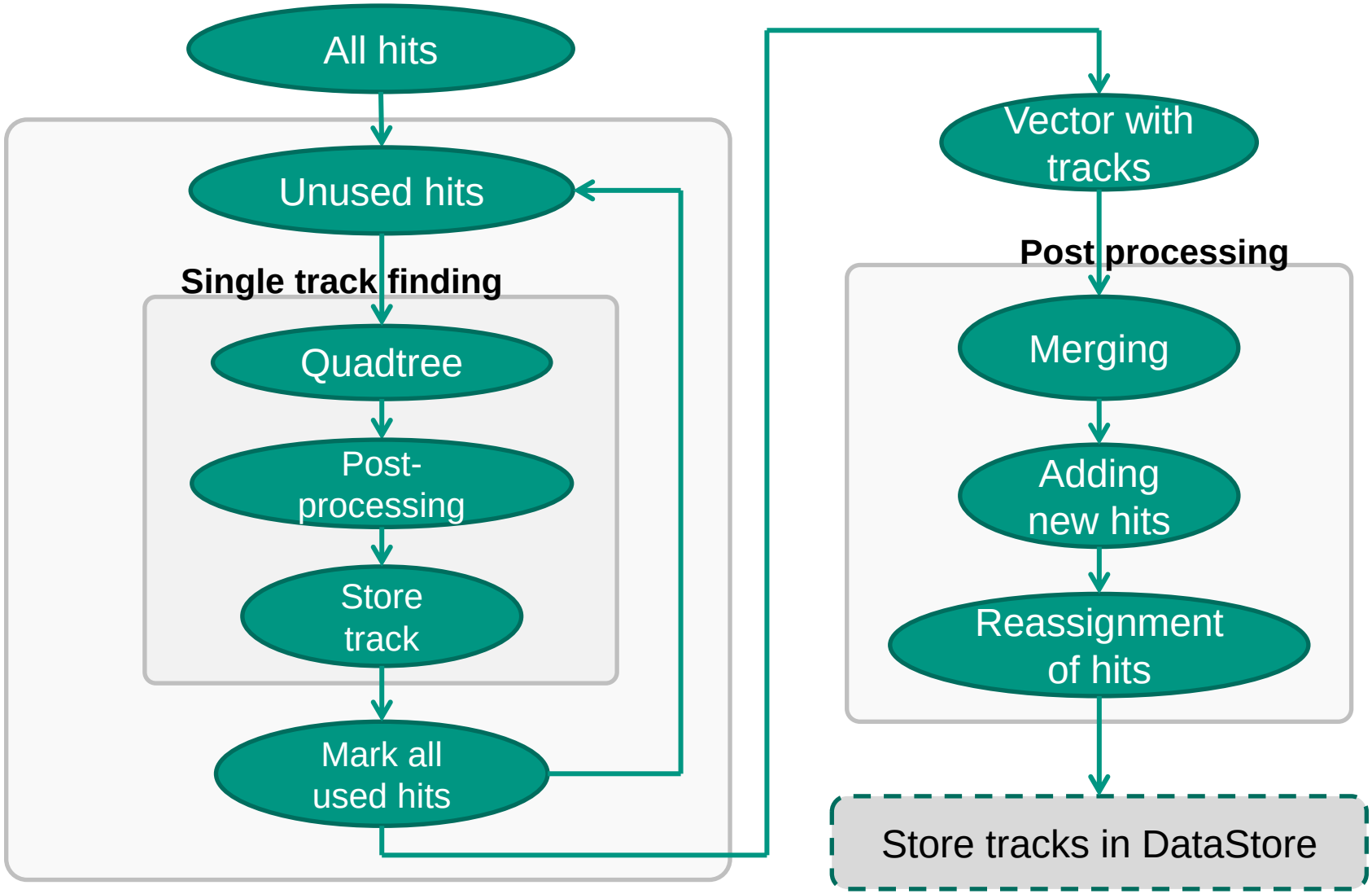
TrackFinderCDCLegendre: overview

Viktor Trusov
01.09.2015 | F2F Meeting in Karlsruhe

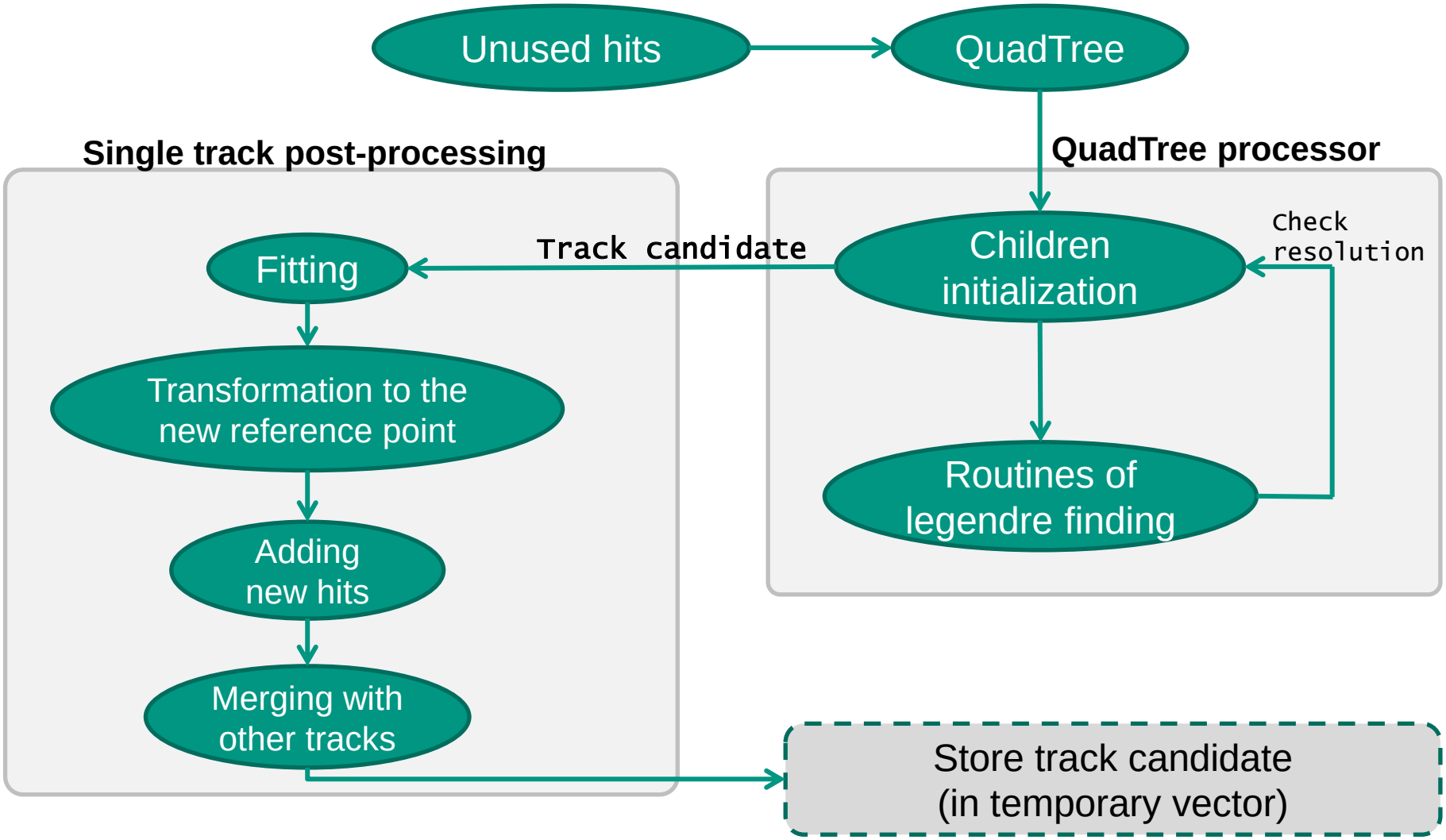
Karlsruhe Institute of Technology (KIT)



Module structure



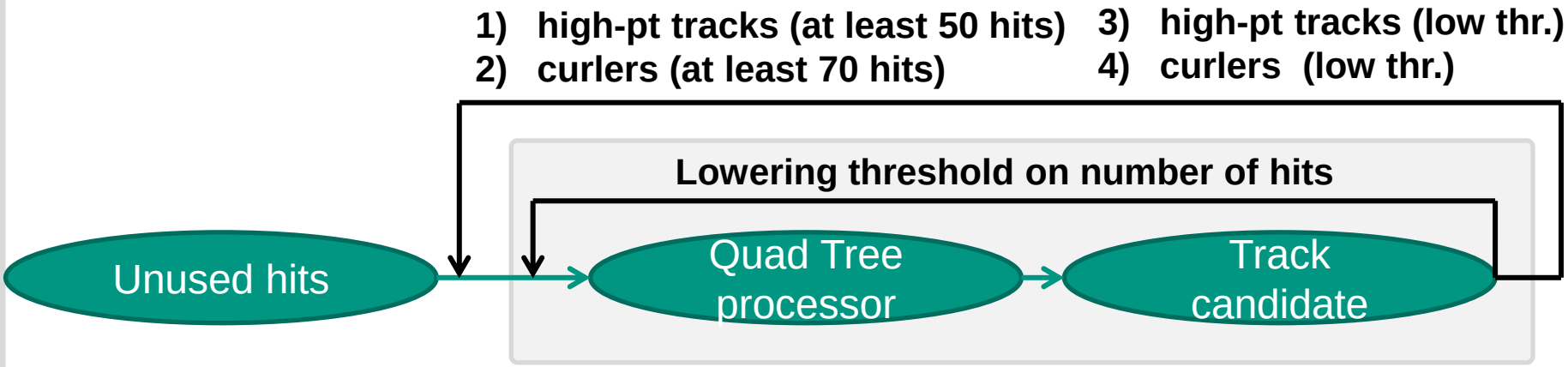
Creation of the single track candidate



Quad tree with quad tree processor

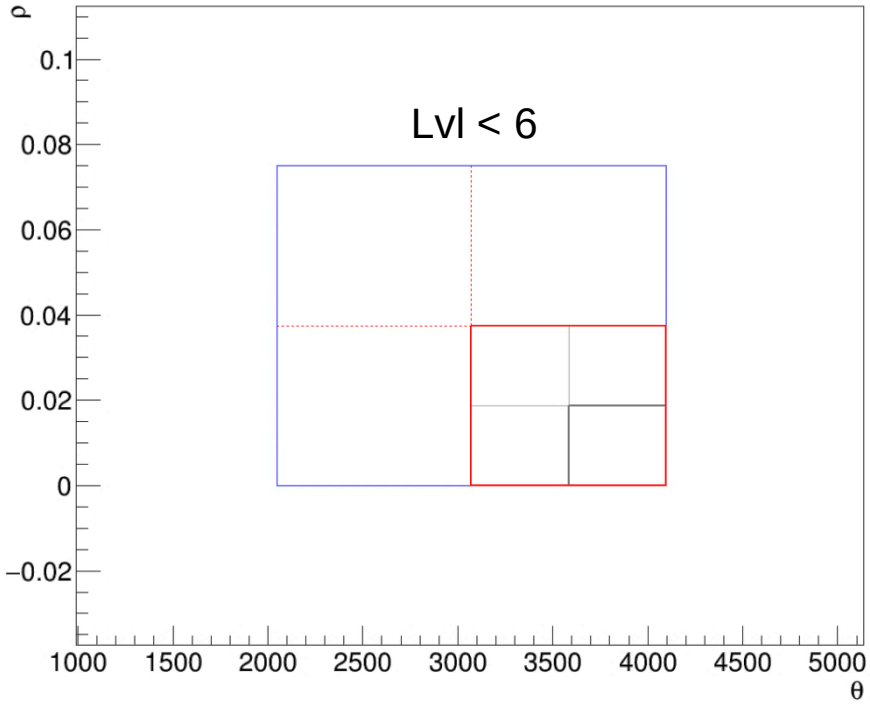
- For finding possible track candidates legendre transformation is used:
  - Similar to the hough transformation
  - Quad tree used as a data-structure
    - Uses quadtree processor for setting search rules
  - Reliable for finding tracks coming from IP

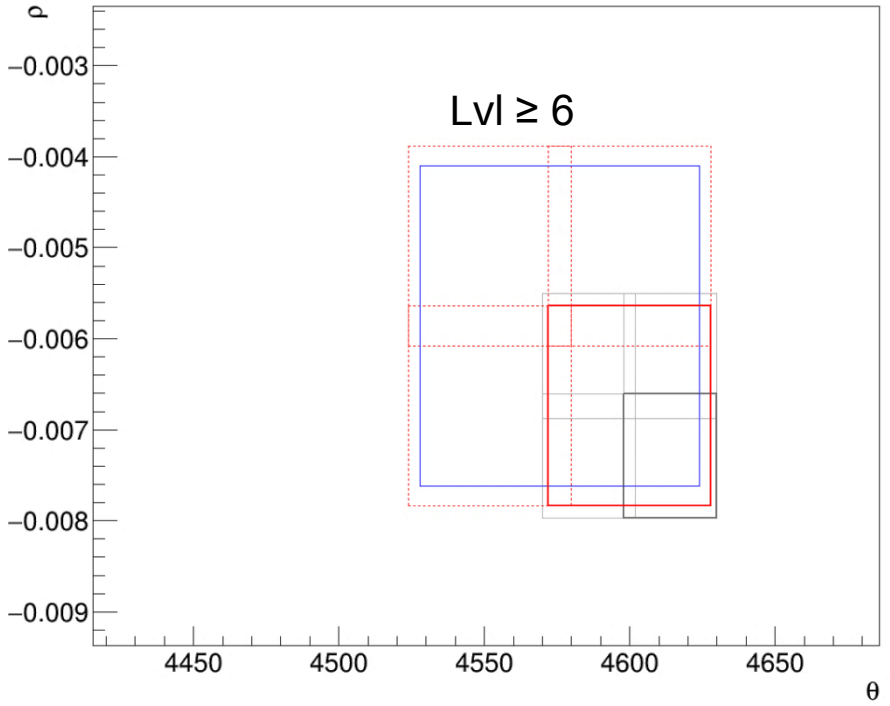
- Few passes of quad tree search with different setting are performed:



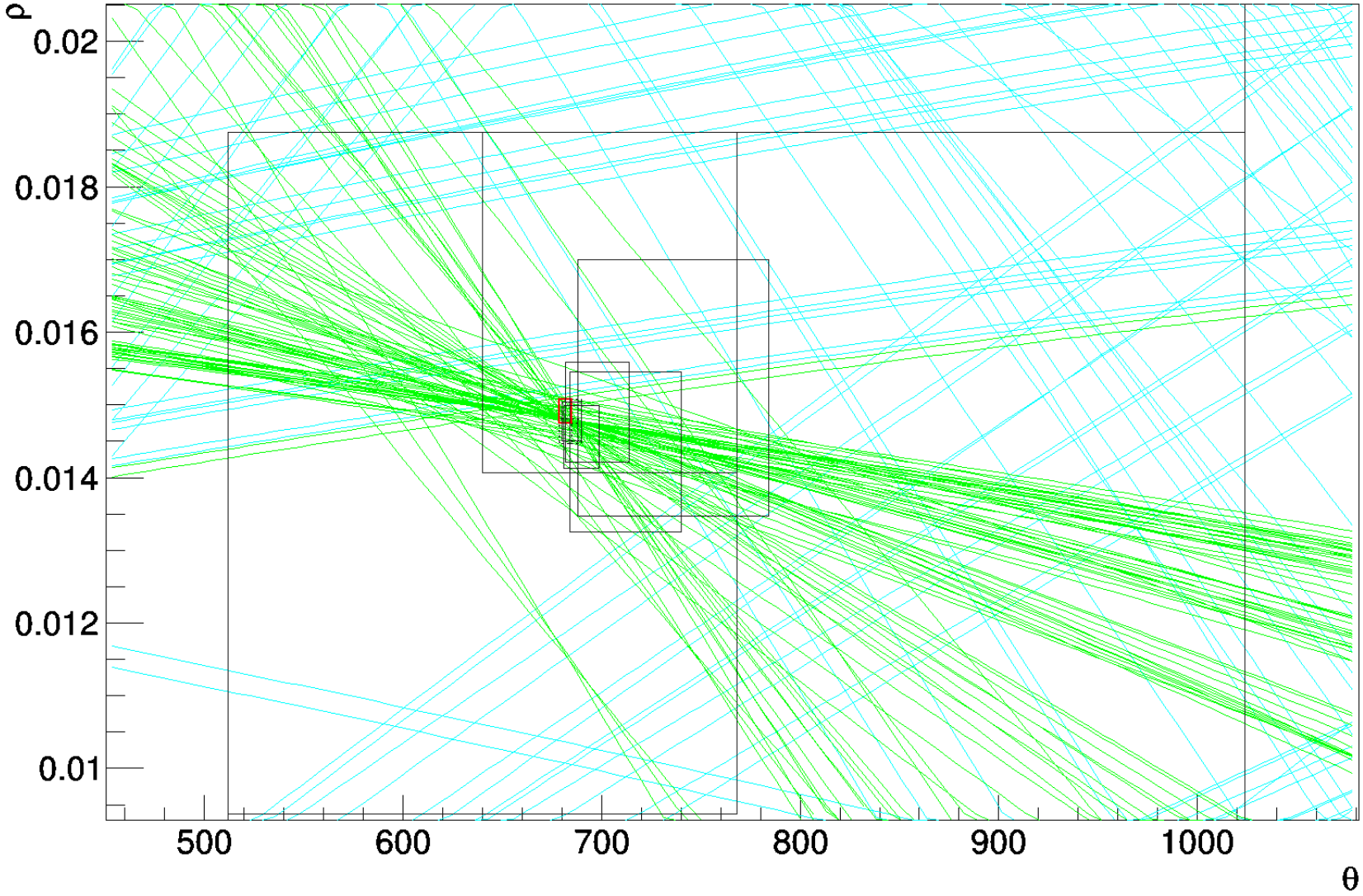
Sliding bins

- For better performance of quad tree search sliding bins were introduced:
  - Allows to decrease border effects
  - Bins (quad tree nodes) are splitted more “smoothly”
  - Used for bins on lvl ≥ 6



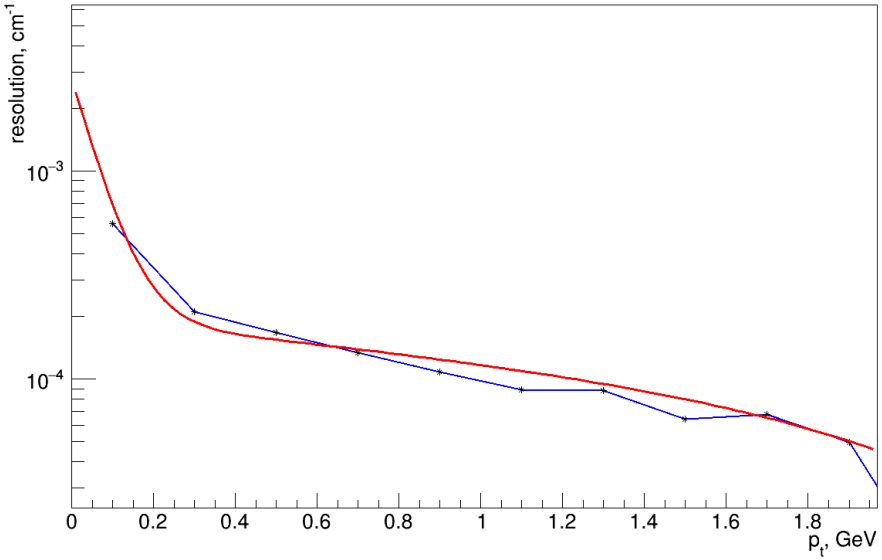


Sliding bins – example



Maxlevel resolution

- Optimal deepness of the quad tree is 13
  - But for low-pt region bins should be more coarse due to material effects
- Instead of setting max. level of the quad tree resolution was introduced
  - Defines size of the lowest bin
  - Implicitly takes into account all effects that bring inefficiencies
  - Estimated using MC simulation
- Different functions are used for different cases
  - Tracks from IP
  - Tracks from non-IP
  - etc

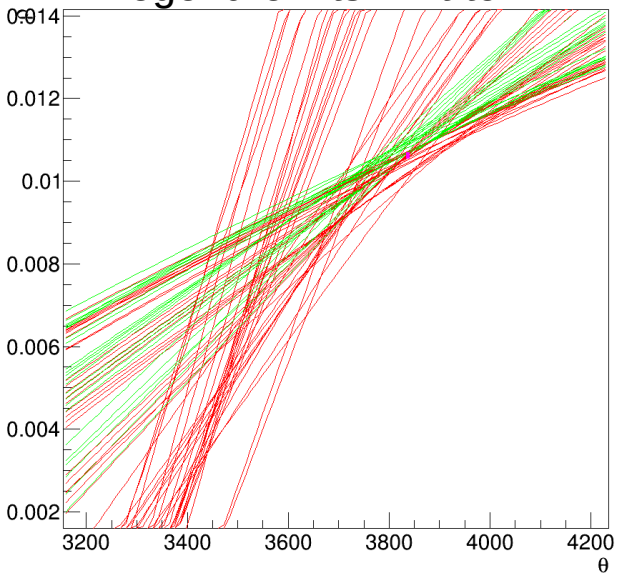


Fitting and transformation to the new reference point

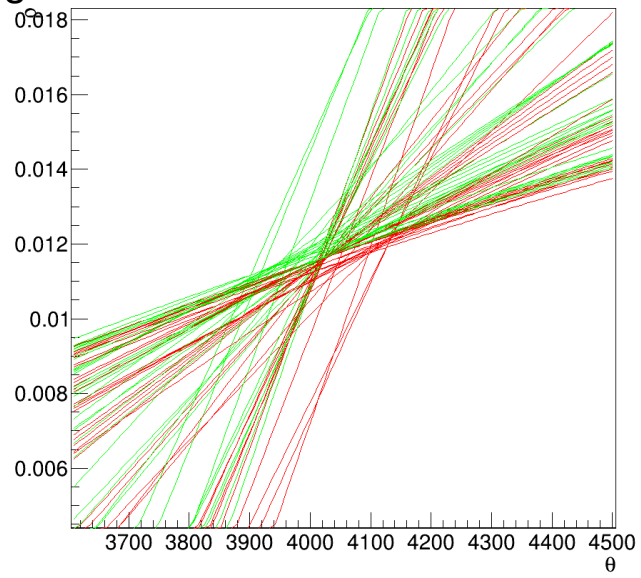
- Track fitting:
  - Karimaki circular fit is used
  - Gives reliable results
- As track is successfully fitted POCA is taken as reference point
  - Conformal transformation performed with respect to new ref. point
  - Single quad tree node created and filled with updated hits

Single track;
$d_0 = 3cm$

Legendre hits w.r.t to IP



Legendre hits w.r.t to new reference point



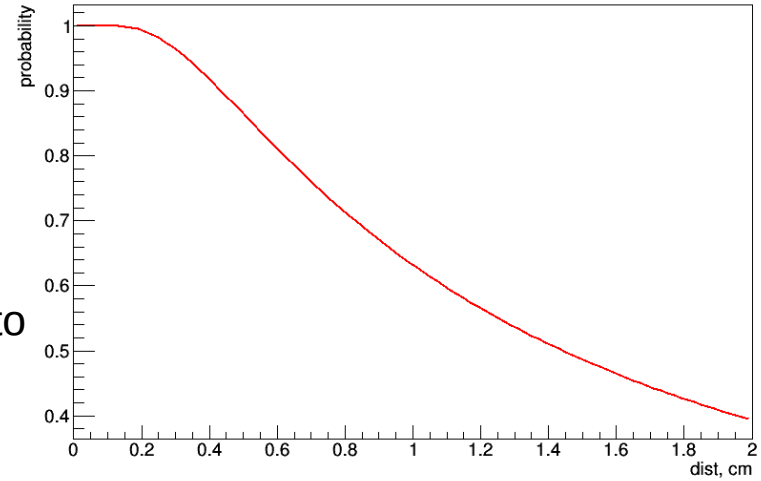
Hits assignment/rejection

- After successful trajectory determination:
  - New hits are assigned (basing on the distance to the trajectory)
  - Bad hits are removed
    - Back-to-back tracks are splitted
    - Hits which are far away from the trajectory are rejected

- Before refactoring:
  - Assignment probability value was used:

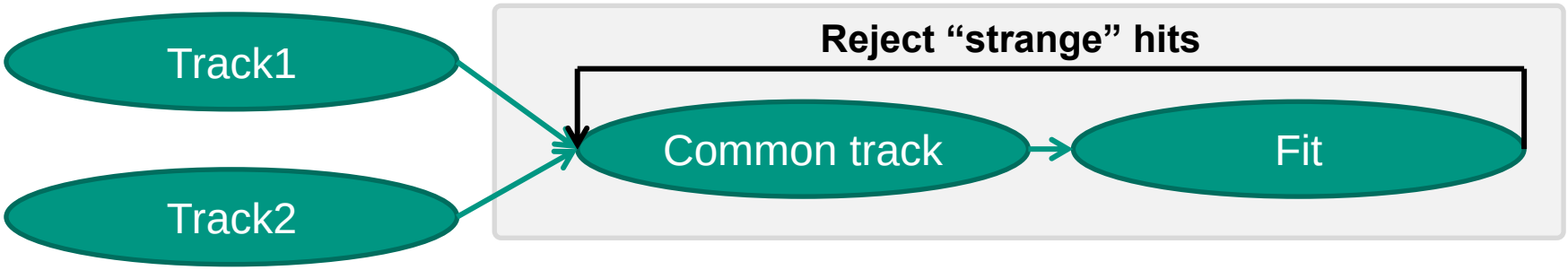
$$1 - \exp 1/d,$$

    - where d – distance from hit to trajectory
    - Hits with probability >0.8 were assigned to the track
- After refactoring:
  - Hits assigned to the track if $d < 0.2\ cm$



Track merging

- Merging of tracks are based on comparing of chi2 of tracks before merging and after
  - Each track checked for compatibility with others



  - Hits were rejected in few stages by lowering threshold on distance to the common trajectory
  - tracks couldn’t be merged:
    - if after few iteration number of hits dramatically decreased
    - if resulting chi2 is not satisfactory
  - For each track best candidate for merging selected the best one

Module refactoring

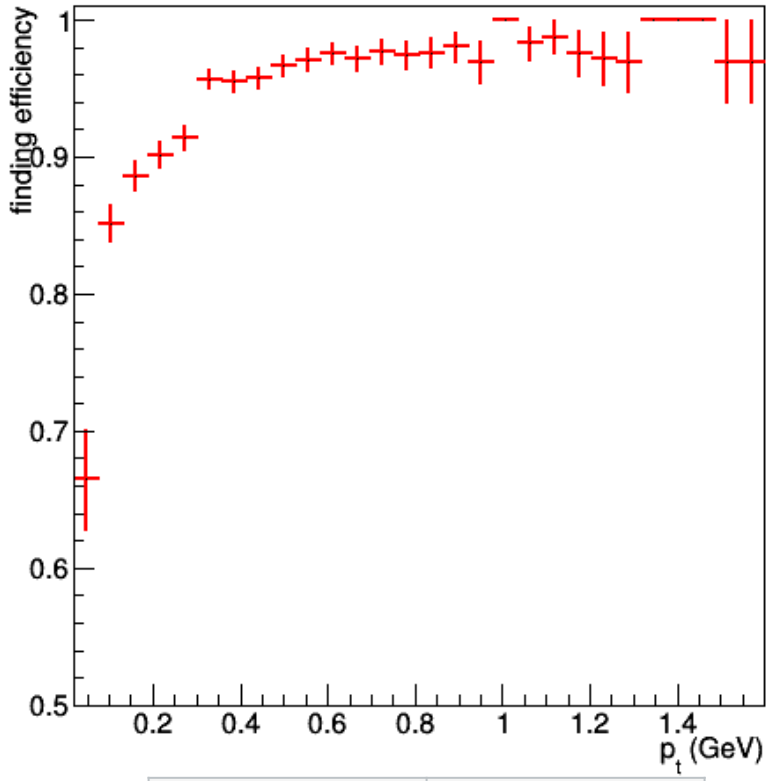
- Currently GlobalFinder is under heavy refactoring:
  - Moving from `TrackCandidate` to `CDCTrack` class
  - Rejection of `TrackHit` class and moving to `CDCRecoHit3D`
    - still `QuadTreeHitWrapper` is used as `CDCWireHit` wrapper and dataobject in quadtree structure
  - Reworked hit assignment procedures

- There are problems with fitting (and as result – merging):
  - The chi2 of the fits are unreasonably high – by orders of magnitude
  - May be as result of underestimated uncertainties of the hits

Efficiency

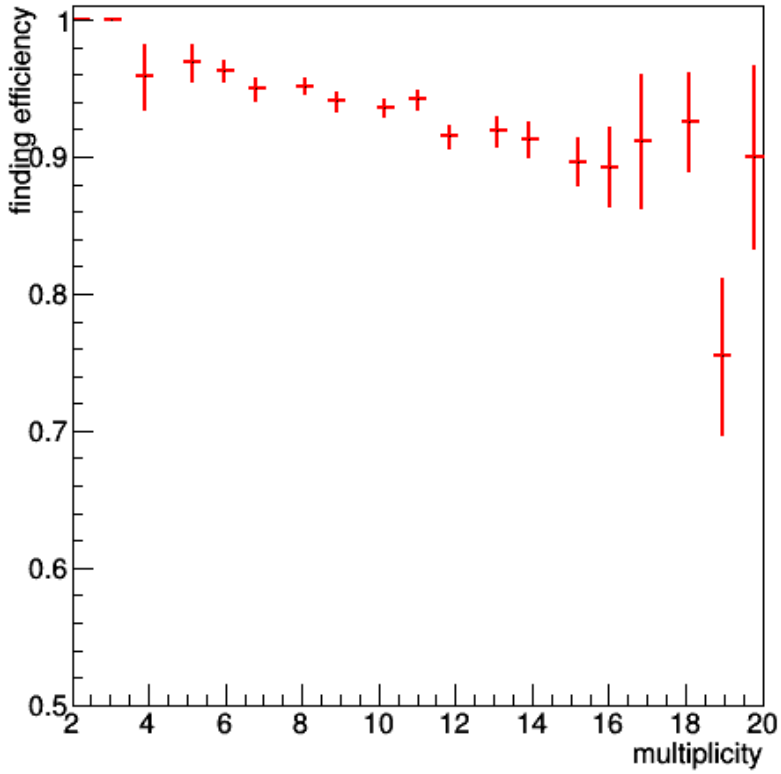
- Primary particles
- Generic $B\bar{B}$, without background

Track Pt



Fake rate	39%
Clone rate	9.3%

Event multiplicity



Conclusion

- Global finder shows reliable result in all pt regions
- Still affected by the fake tracks

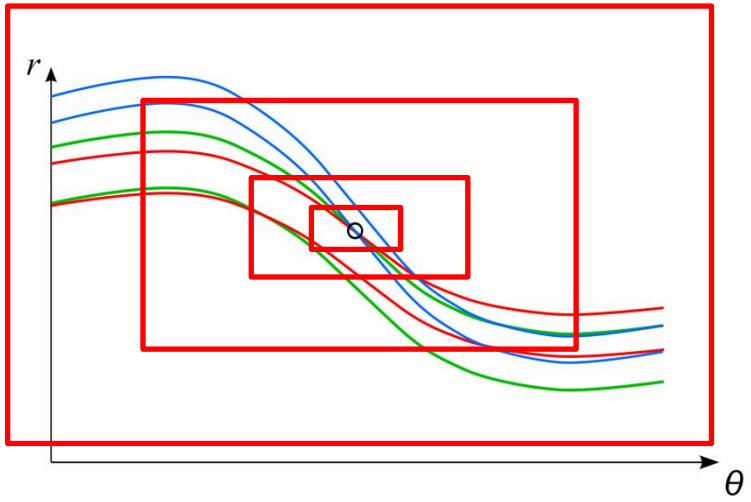
- Refactoring brings opportunity to introduce improvements into the module
  - Decrease fake rate
  - Decrease CPU time
  - Etc

- Plans:
  - Finish refactoring
  - Improve quality criteria of the tracks
  - Make robust merging

Backup

How to measure resolution?

- Using particle gun single tracks were generated
  - $p_t < 2\ GeV$
  - $d0$ – normal distribution with $\sigma = 3\ cm$
- Using parameters of the track single QuadTree node was created
  - QuadTree node centered
- Boundaries of the node taken as whole legendre phase space
- If all generated hits could belong to the node – reduce its size by factor 2
- Repeat until desired number of hits still belongs to the node



- Final size of node characterizes resolution for track with given parameters

Resolution – *pt*

- Resolution also depends on p_t
  - For lower p_t it's higher (as expected)
  - Unexpected: bump around 0.8 GeV