Track reconstruction with the pixel and the full CMS tracker

Teddy Todorov

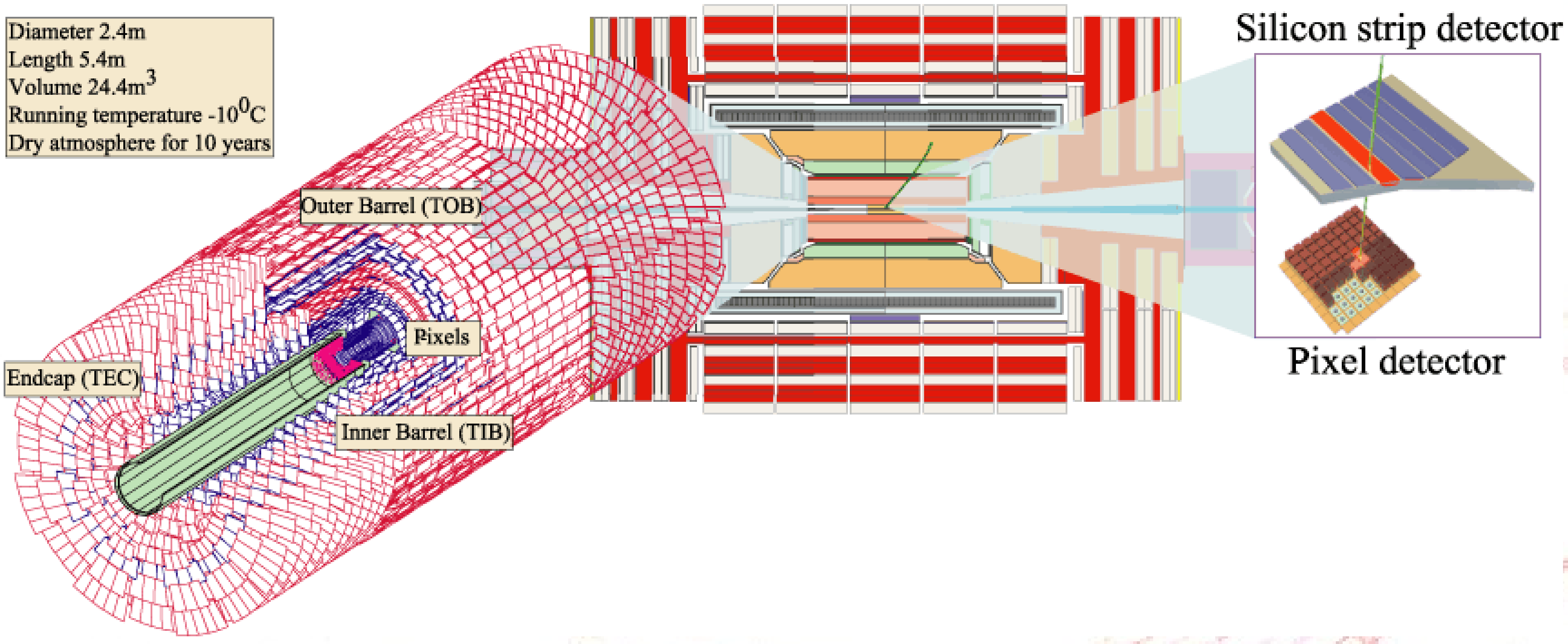
IReS, Strasbourg / CERN

Time05

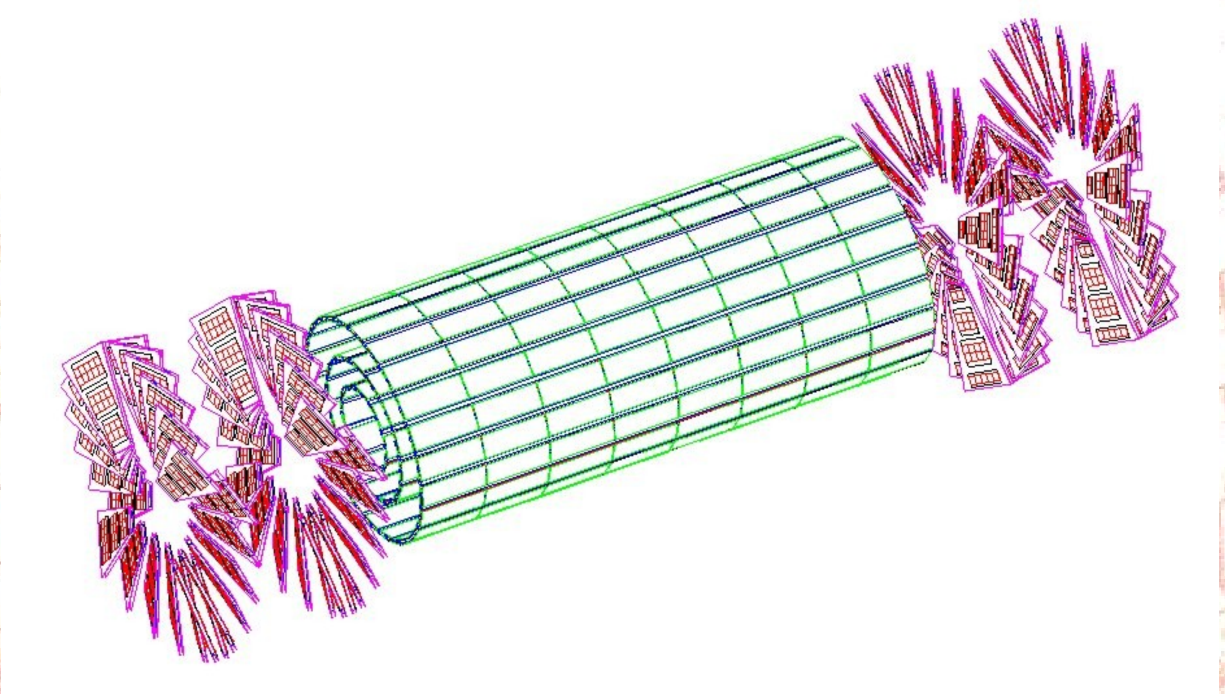
5/10/2005

The CMS tracker

- All-silicon tracker
  - ~3 pixel hits per high momentum track
  - ~10 strip hits, 4 double-sided and 6 single-sided



The pixel detector

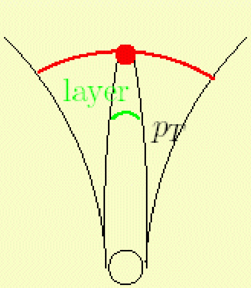


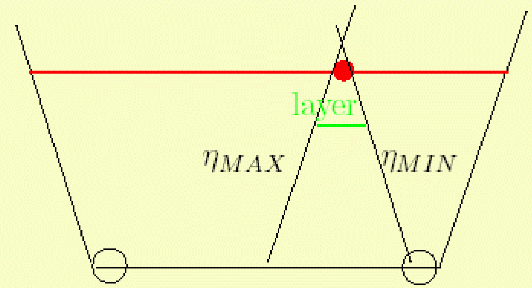
- Three barrel layers, at 4, 7.and 10 cm radius
- two endcap disks
  - rotated sensors to improve resolution via Lorentz drift charge sharing
- pixel size 100x150 μ, 66 million channels
- eta coverage up to ~2.5

Pixel triplets

- Geometrical search for triplets of pixel hits
- Constraints:
  - compatible with interaction region (cylindrical)
    - not the beam spot!
    - size defined by physics analysis
    - typical size is 1- 2 mm in radius, ±15 cm in Z
  - Transverse momentum above some cut
  - Global case: eta within tracker acceptance
  - Regional case: eta and phi cuts
- Parametrized multiple scattering, no energy loss
- Highly optimized implementation

The hits from layers are accessed more than once. Since the φ constraint is more predictive than r/z the hits are kept in φ-sorted cache.

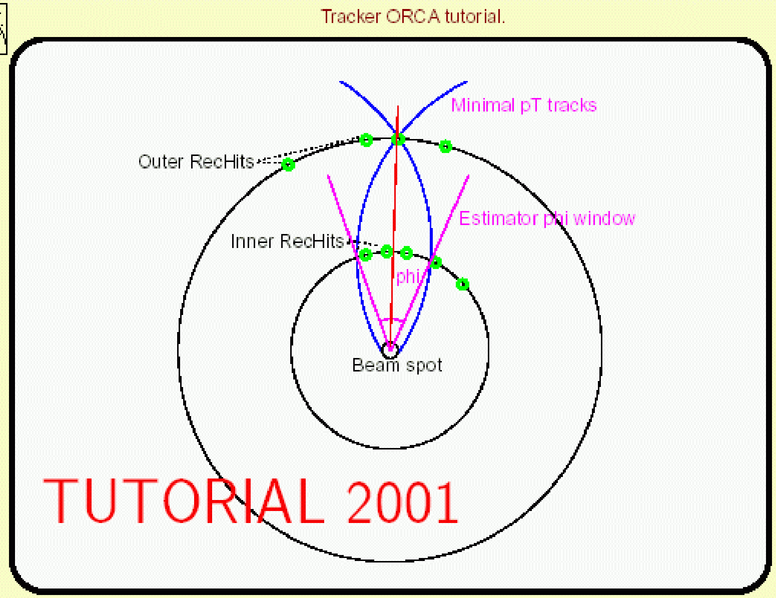




- using the analytical prediction for φ the STL binary search is used to find hits compatible in phi. No direction constraints are used.
- Each hit is tested against r/z.
  rz constraint from region is used.

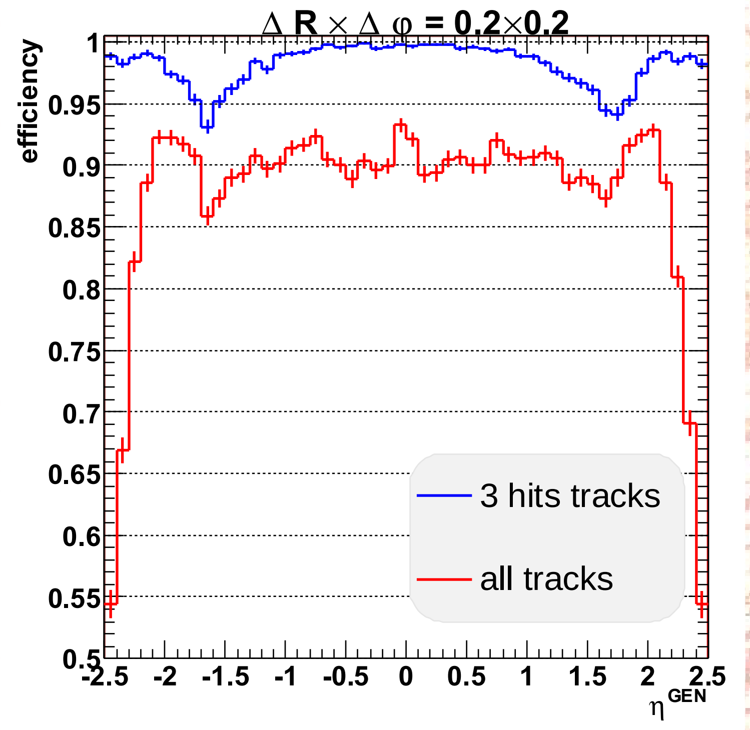
Key points:

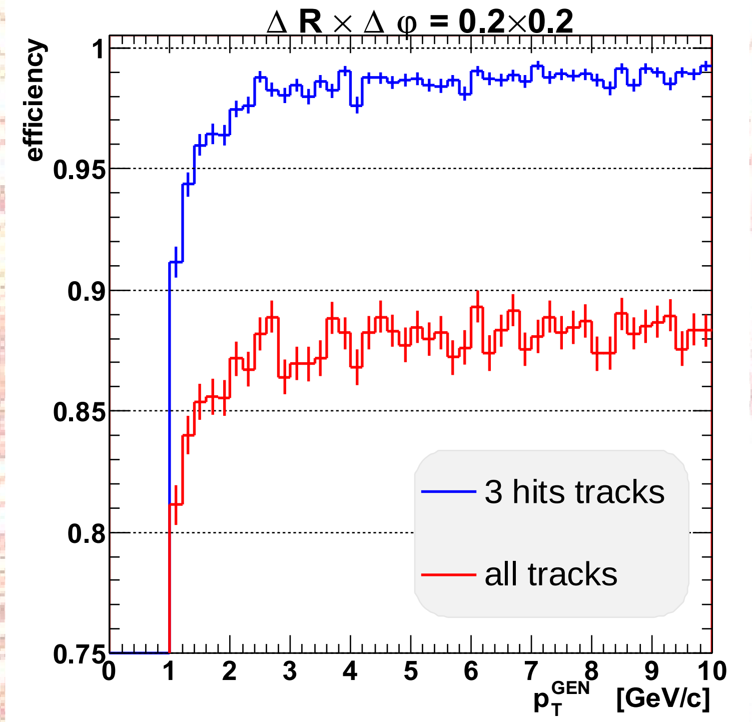
- caching,
- optimal sorting
- fast searching



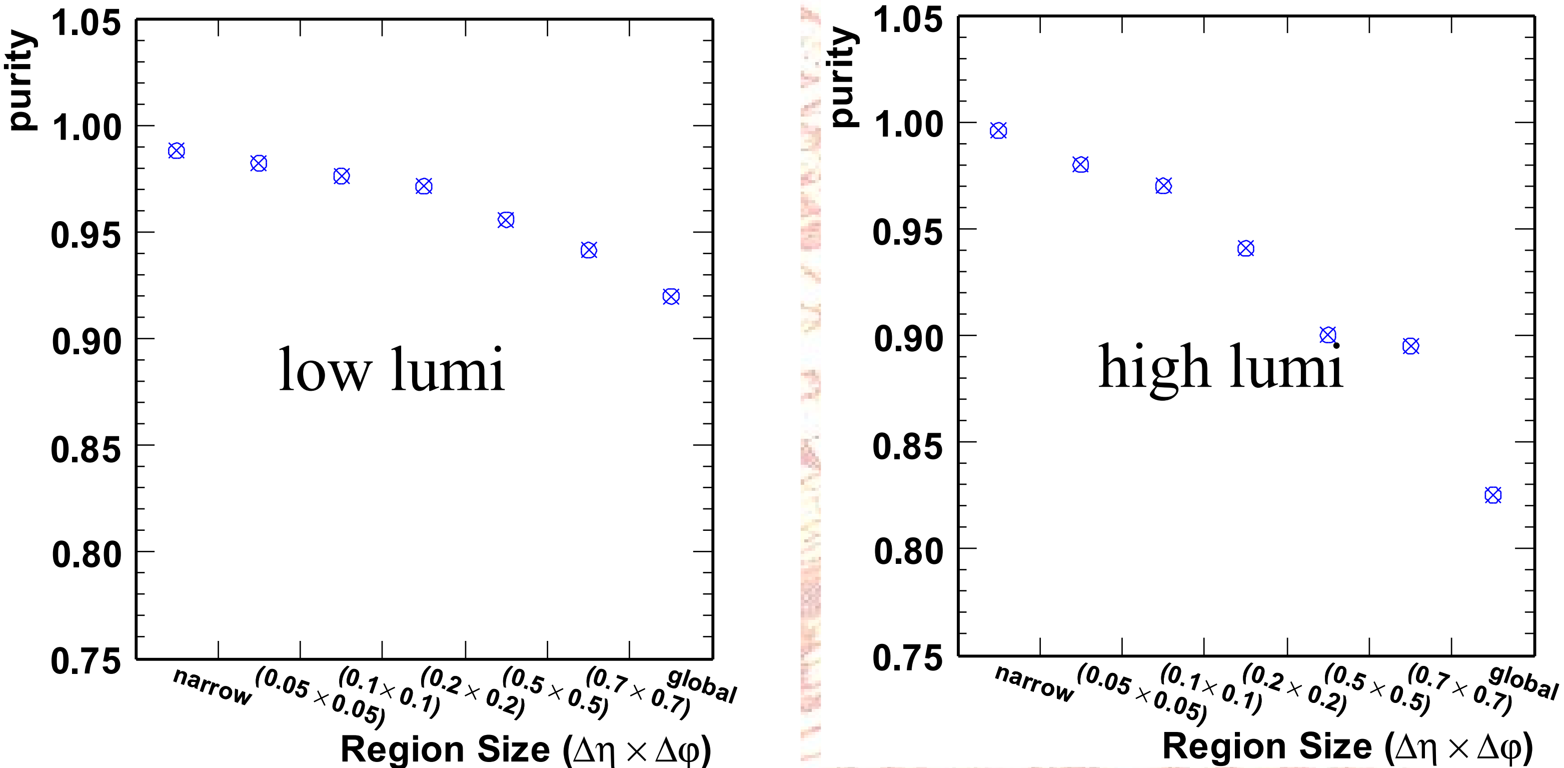
Pixel triplet efficiency

- Blue line for tracks with three pixel hits
- Red line for all tracks
  - difference due to geometrical inefficiencies but excluding readout inefficiencies





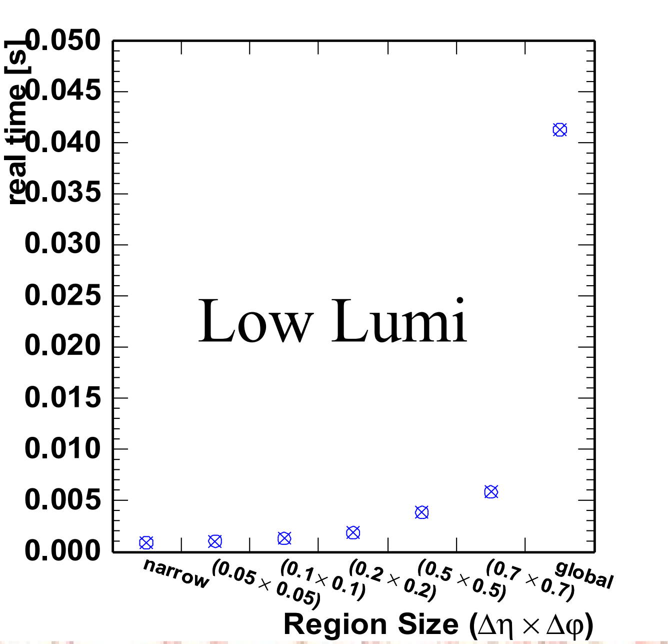
Pixel triplet purity

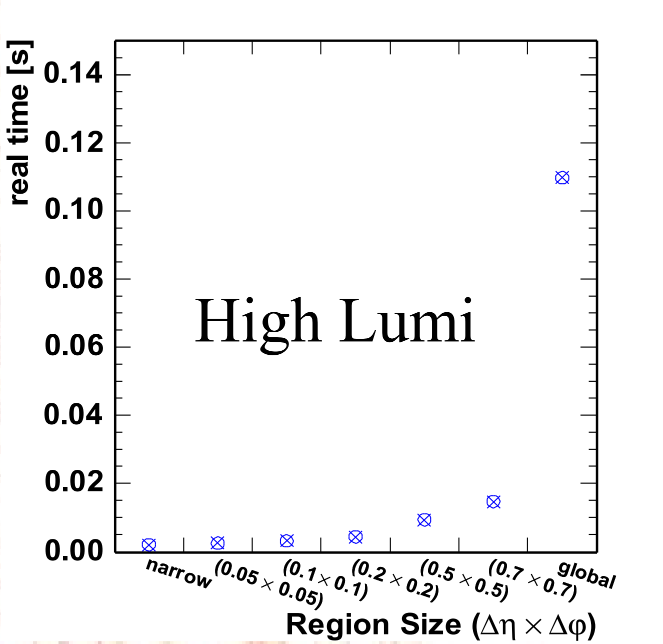


- ratio of pixel triplets from tracks to all pixel triplets
  - just an example, depends on physics channel and region definition

CPU time for pixel triplet finding

- CPU time, on a 2.4 Ghz Xeon CPU



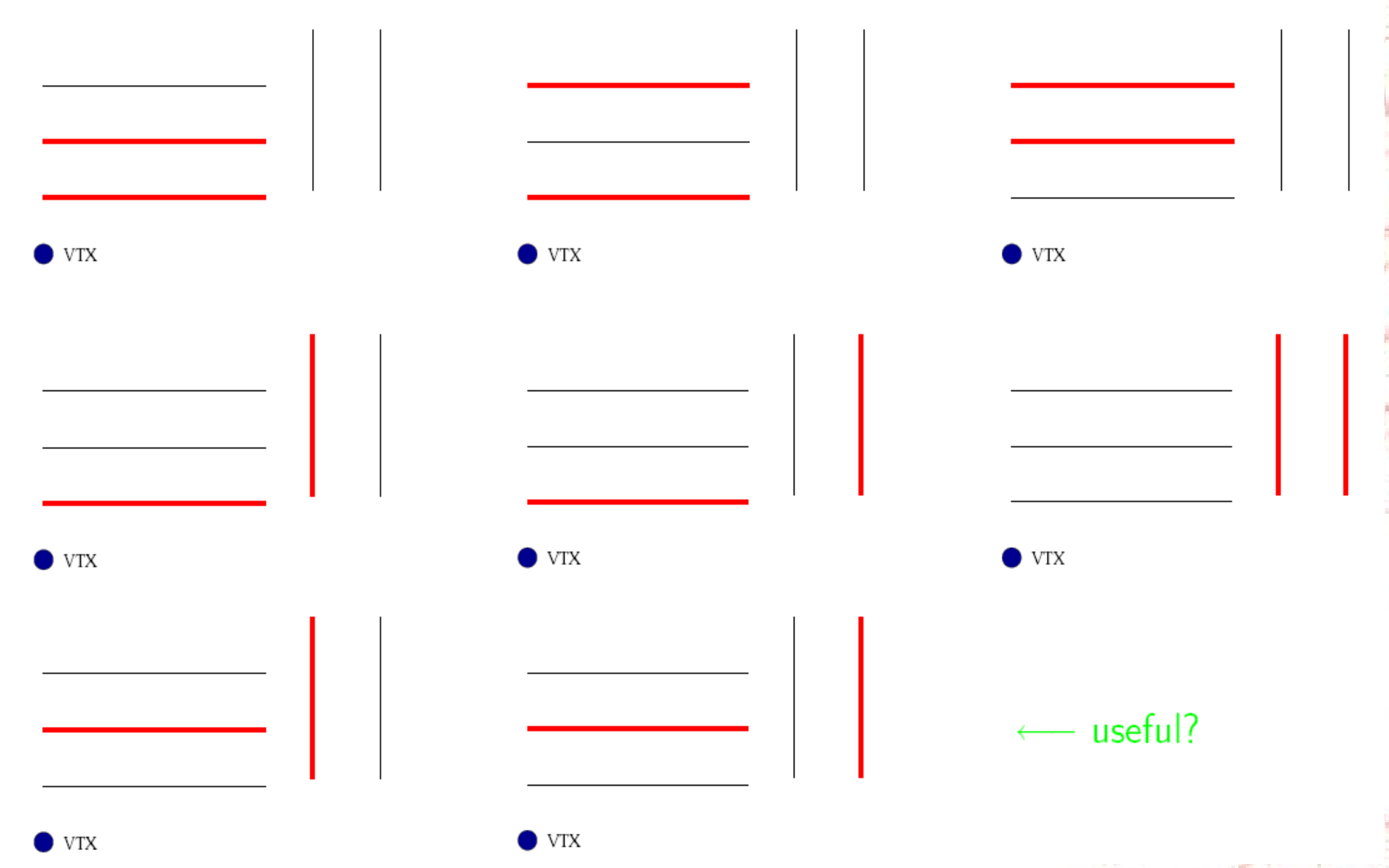


Pixel triplets summary

- Pixel triplets can be reconstructed
  - efficiently (~90%)
  - with good purity (~10% ghosts)
  - very quickly (in a fraction of the HLT time limit)
- The total number of pixel triplets per event is small
  - a few hundred even at high luminosity
- The pixel triplets are ideal seeds for Kalman filter pattern recognition
  - all 5 track parameters well constrained
- Pixel triplets can even be used as tracks
  - in High Level Trigger

The last 10%

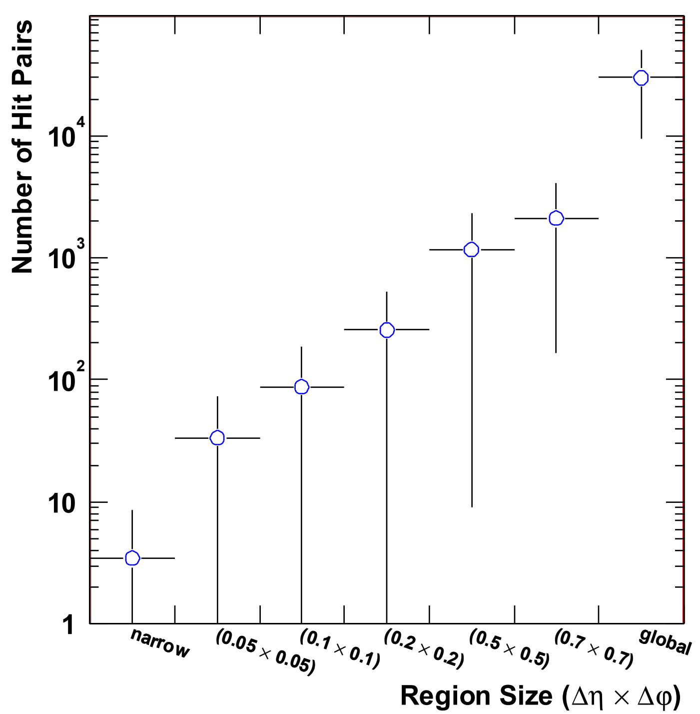
- The last 10% of the tracks take more than 90% of the CPU time!
- Require use of “2 out of 3” pixel layers
-

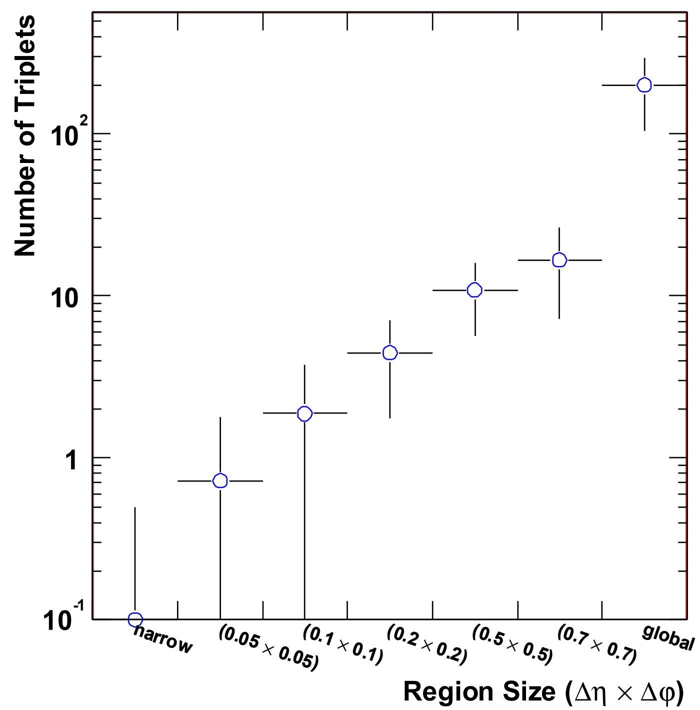


Hit pair combinatorics

- Reconstructing the pixel triplet tracks does not reduce the combinatorial problem for the remaining tracks
  - most of the hits do not come from reconstructible tracks
  - only about 3% of the hit pairs can be removed

- At high luminosity, the number of hit pairs is 20 – 30 thousand
  - about 100 times more than the number of triplets





Hit pair reconstruction

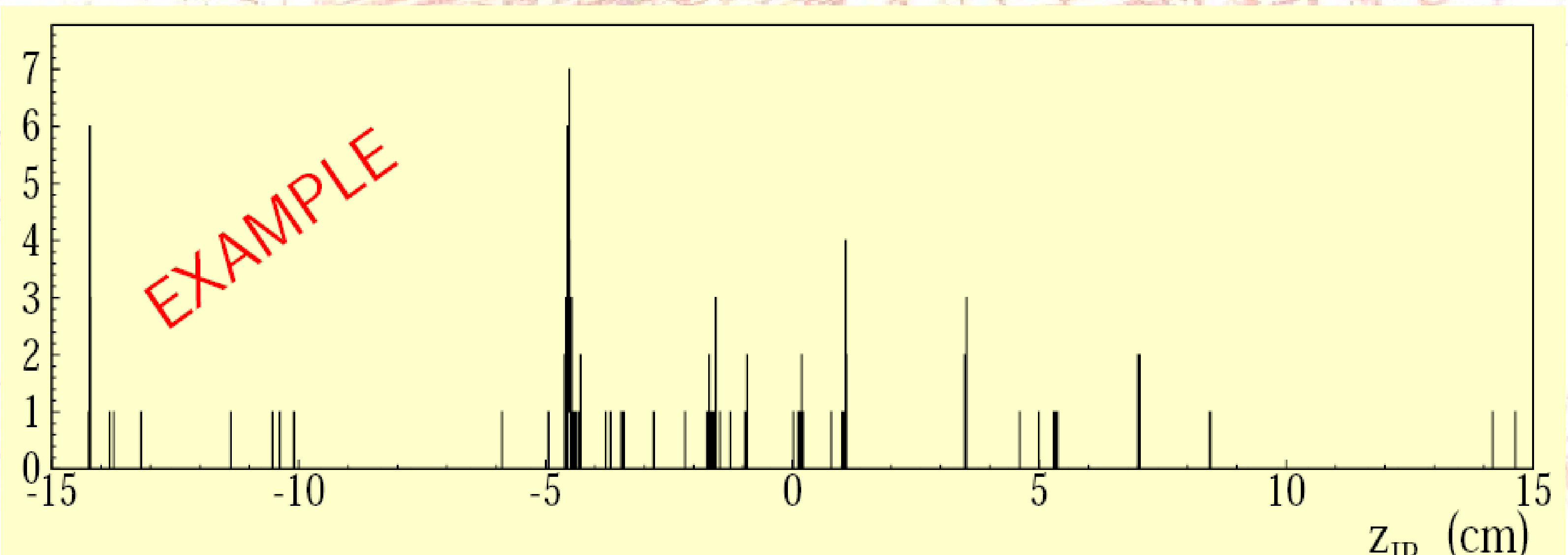
- Very efficient
  - more than 99%, “good enough” for all purposes
- Very low purity
  - of the order of 1% (99% ghosts)
- CPU time similar to triplet reconstruction
- Require additional assumptions (compatibility with interaction region) to constrain all 5 track parameters
  - less precise seed parameters than triplets

Cleaning seeds with primary vertex

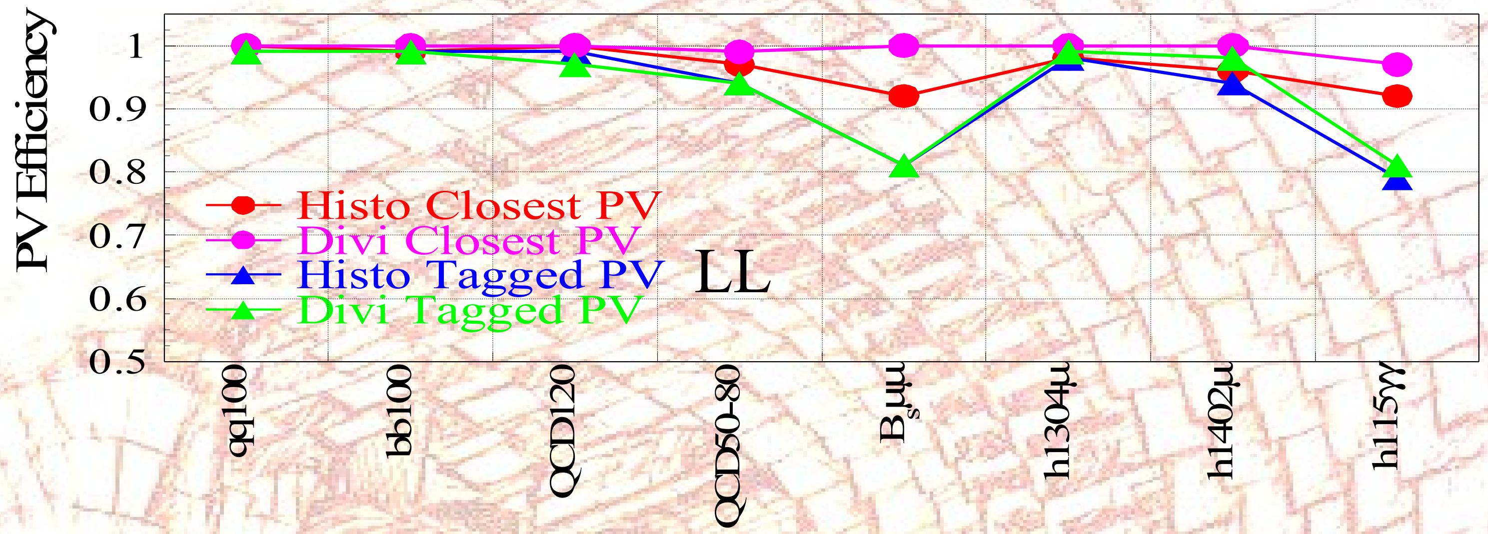
- If the primary vertex is known, the hit pairs not compatible with it can be eliminated
  - for reconstruction of the “trigger” event
  - in some cases reconstruction of tracks from pile-up events may also be required
    - e.g. energy flow
  - Large reduction in number of pairs
    - by a factor of 6 at low lumi, more than a factor of 10 at high lumi
- Primary vertex may be defined by trigger muon, electron, di-muon, etc.
- Primary vertex can be reconstructed before track reconstruction
  - **from pixel triplets!**

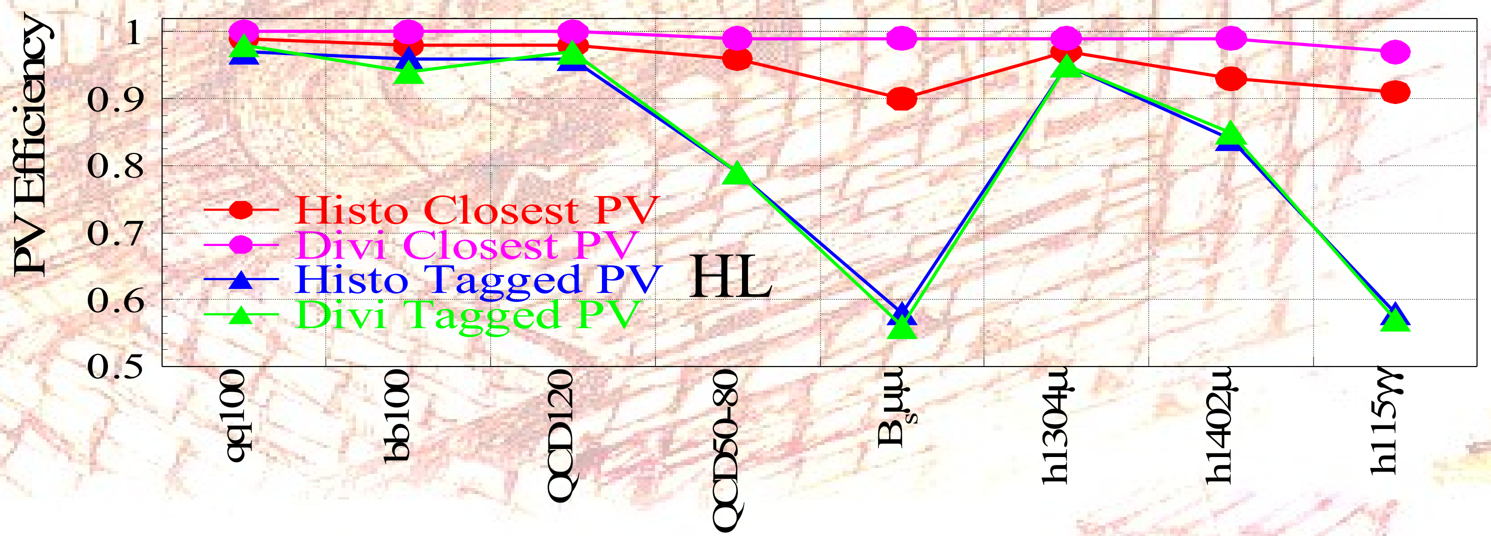
Pixel primary vertex finding

- Pixel triplets cluster (in Z impact parameter) around the primary vertex
- A simple clustering or histograming method in 1D is enough to find the vertices
- Identifying the trigger primary vertex is not always easy
  - depends on the type of the trigger event



pixel PV finding efficiency





Track reconstruction

- Seed generation is only the first stage
- Each seed is followed in a combinatorial Kalman filter
  - "trajectory building"
  - No hit locking, all seeds tracked independently
- Mutually exclusive tracks (sharing large fraction of hits) must be "cleaned"
  - based on normalized χ^2, with a penalty for missed hits
- A final refit with smoothing removes potential bias from the seed

Combinatorial growth

- Most tricky problem: if not limited, leads to exponential increase of number of candidate trajectories
- All candidate trajectories (from a single seed) are grown in parallel, one layer at a time
- after the inclusion of measurements from each layer the total number of candidates is limited to a small value
  - e.g. 5
    - this value is within 1 or 2 per mils of the asymptotic efficiency
  - This parameter allows tuning of CPU versus efficiency
    - for HLT tracking the limit is 1

Example of hit combinatorics

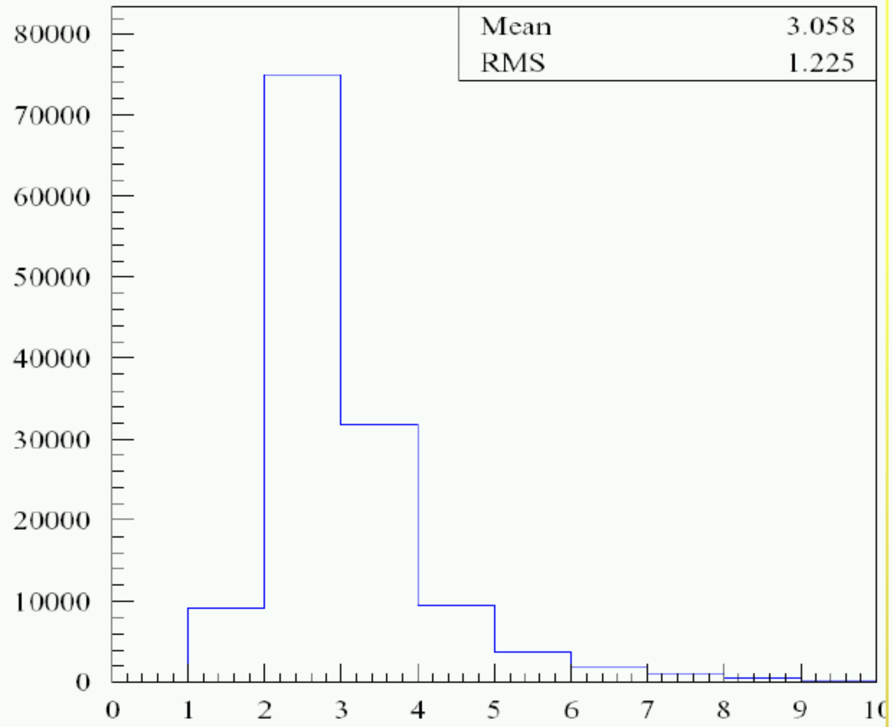


Figure 14: *Number of compatible hits found on TIB layer 1 for each trajectory candidate when leaving Barrel Pixel layer 3 for 100 GeV b jets without pile up.*

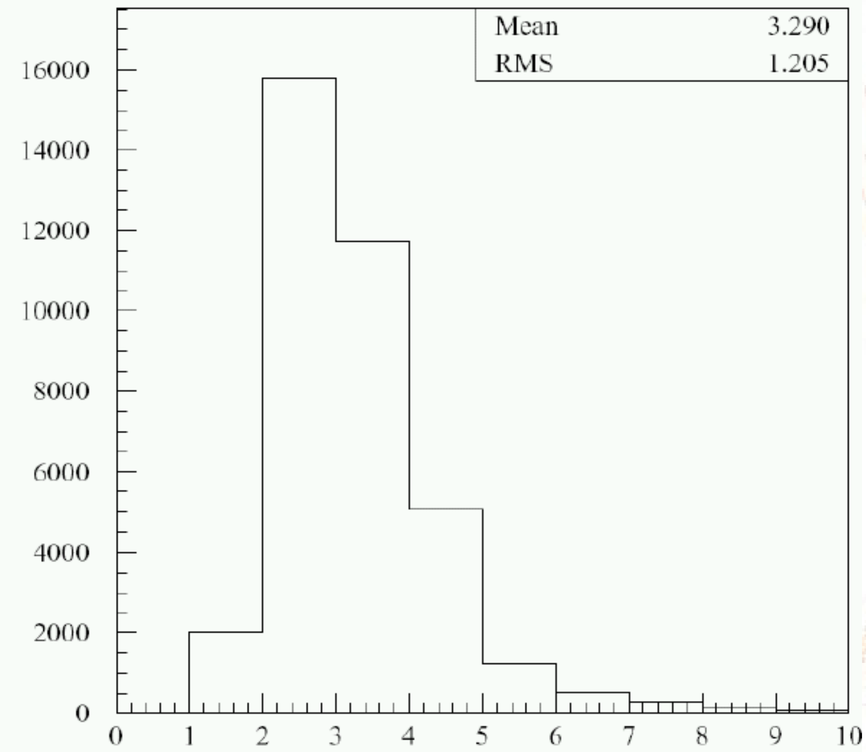
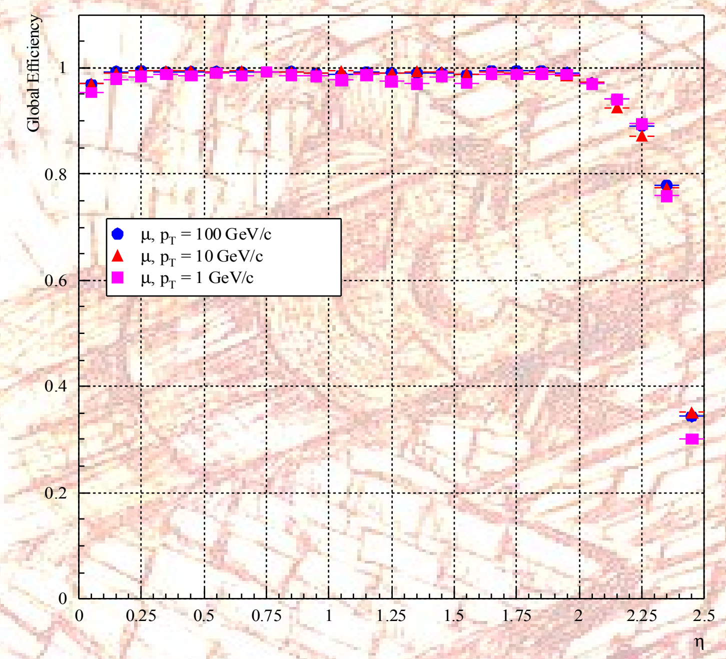
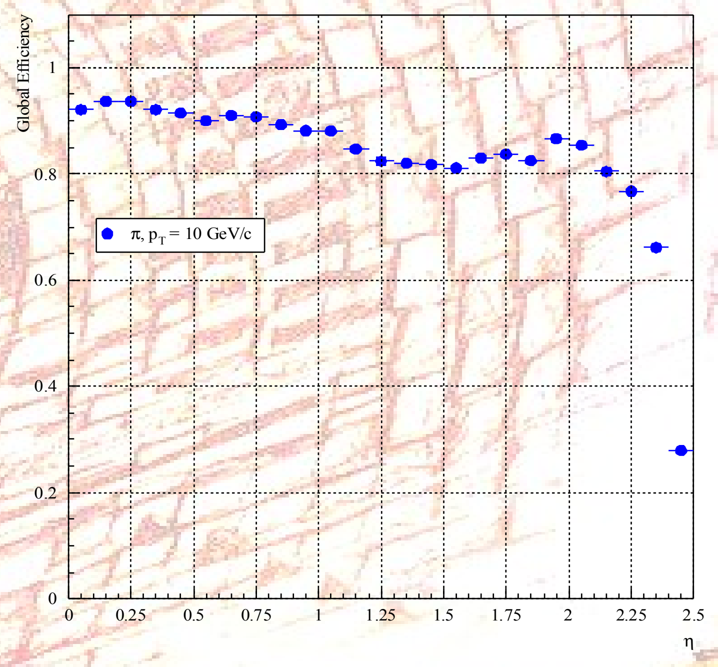


Figure 21: *Number of compatible hits for each trajectory candidate when leaving the Forward Pixel disk 2 (100 GeV b jets without pile up).*

Track reconstruction efficiency

- Efficiency is limited by hadronic interactions
  - between 10% and 20% of the pions (depending on momentum and eta) disappear before leaving 8 hits!



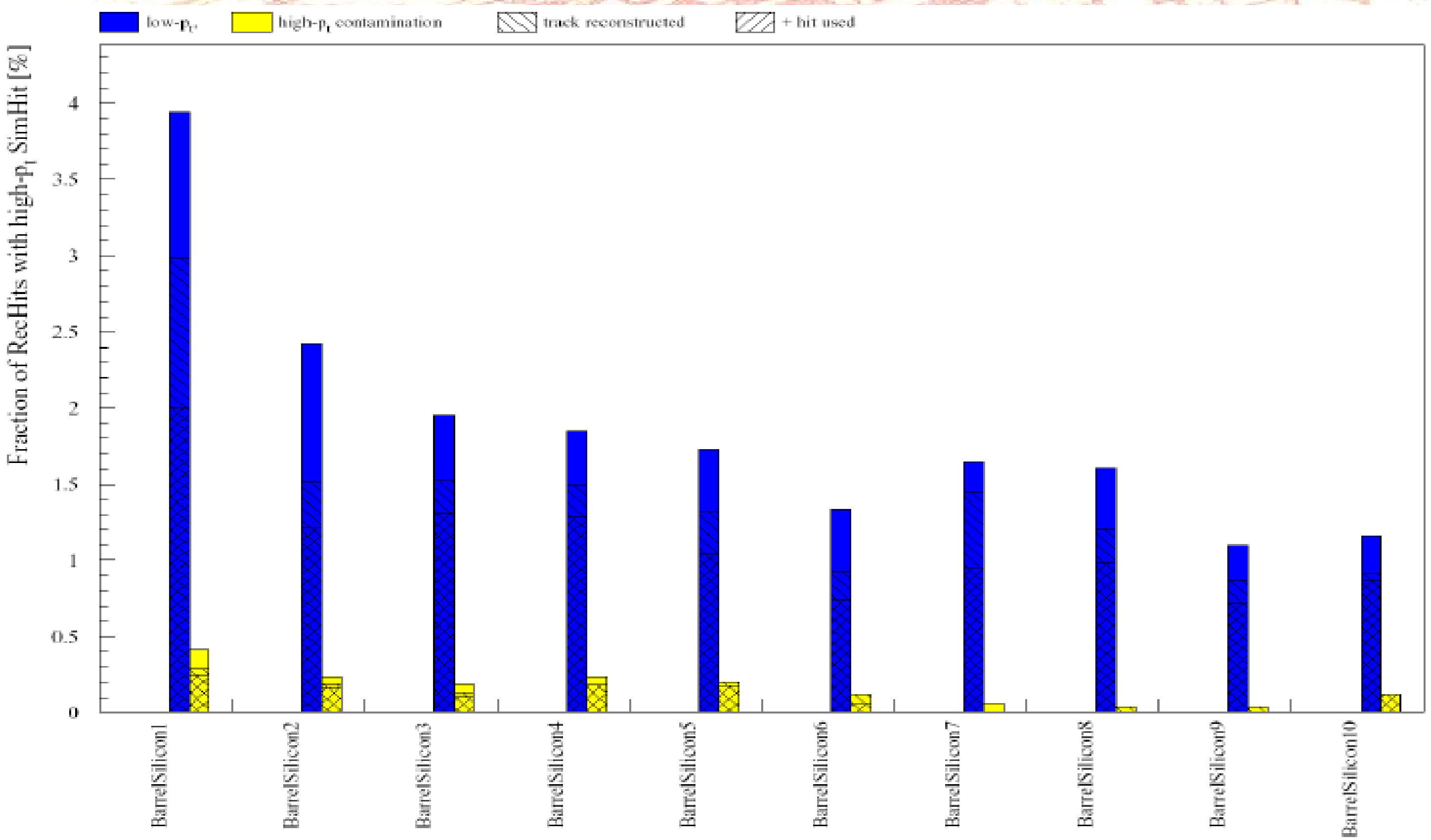


Efficiency in jets

- The efficiency in dense environment (jet core) is close (within few percent) to the single particle efficiency
  - so close that the differences are not yet quantified
  - Not very surprising: a combinatorial search should find the correct combination of hits
    - among other combinations
    - if the hits are not affected

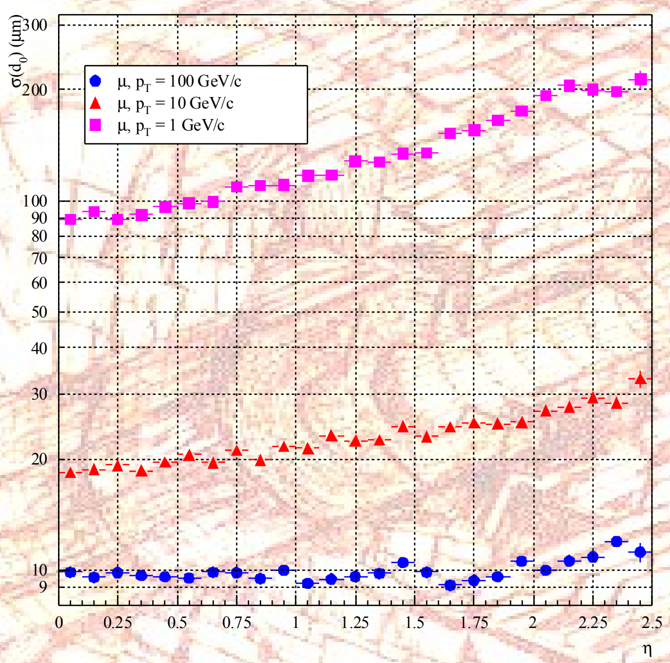
Hit contamination

- For 100 GeV Pt b-jets with high lumi pile-up
  - High Pt (above 1 GeV) contamination in yellow
  - Low Pt contamination in blue

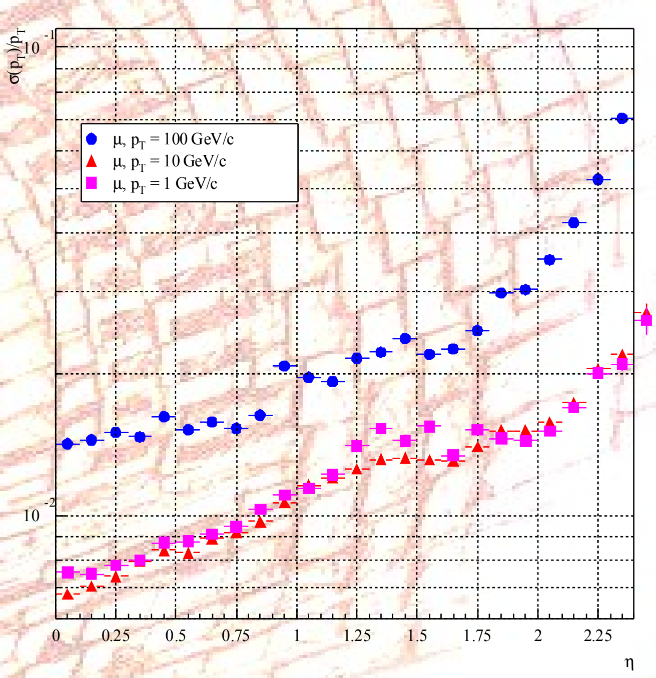


Some resolutions

Transverse impact parameter



transverse momentum



Dense environment?

- If a 100 GeV jet core at high luminosity is not a dense environment for the CMS tracker, what is?
  - a 200 GeV jet is significantly denser
  - hardest case studied for proton-proton: three-prong τ decays
- Is the combinatorial Kalman filter sufficient?
- Implemented and studied
  - Deterministic annealing filter
    - soft assignment of hits to a single track
  - Multi track filter
    - soft assignment of hits to several tracks simultaneously

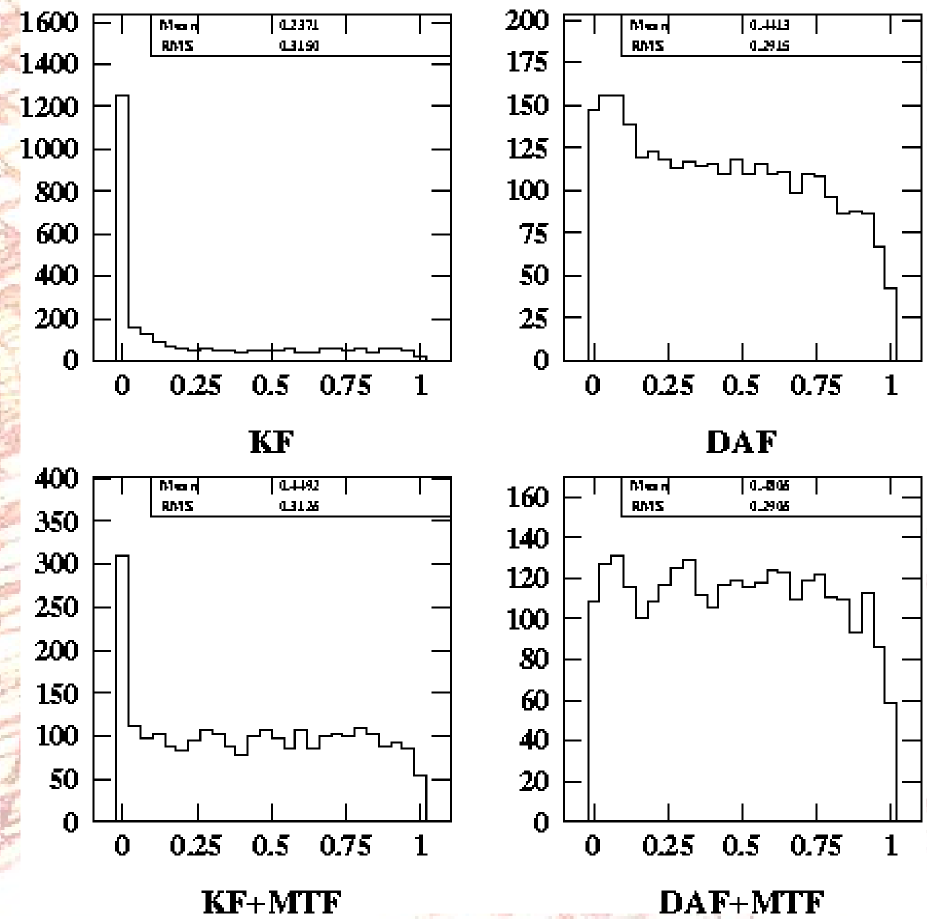
Adaptive filters: the DAF and the MTF

Reconstruction of π tracks from the decay of high-p_T τ:

$H^0 \rightarrow \tau^+\tau^-$, $m(H^0) = 500$ GeV/c^2

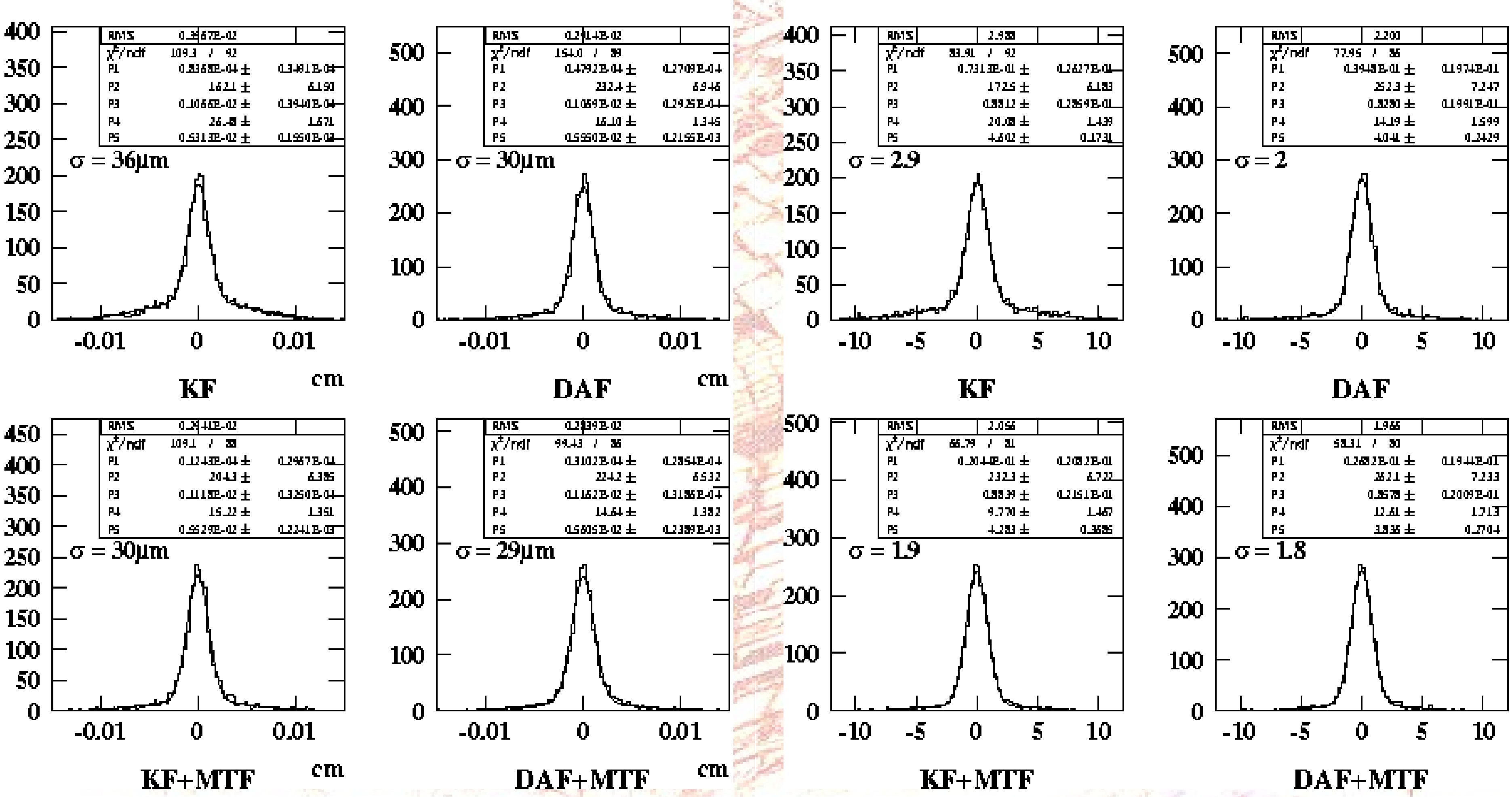
- KF: Kalman Filter alone
- DAF: DAF with seed from KF
- KF+MTF: MTF tracks, seeded with KF tracks
- DAF+MTF: MTF tracks, seeded with DAF tracks

χ^2 probability



Adaptive filters: the DAF and the MTF

Transverse IP resolution | Transverse IP pull



Little improvement with the MTF over the DAF

Conclusions

- The CMS tracker allows for very robust tracking up to the LHC design luminosity.
  - Caveat: this is a Monte Carlo study with ideally aligned and calibrated tracker
  - effects of misalignment presented tomorrow (N. de Filippis)
- The pixel detector has extensive capabilities
  - for stand-alone reconstruction
  - for seeding the track reconstruction
- Advanced track algorithms (DAF...) bring measurable improvements in very dense environments