Track reconstruction in CMS

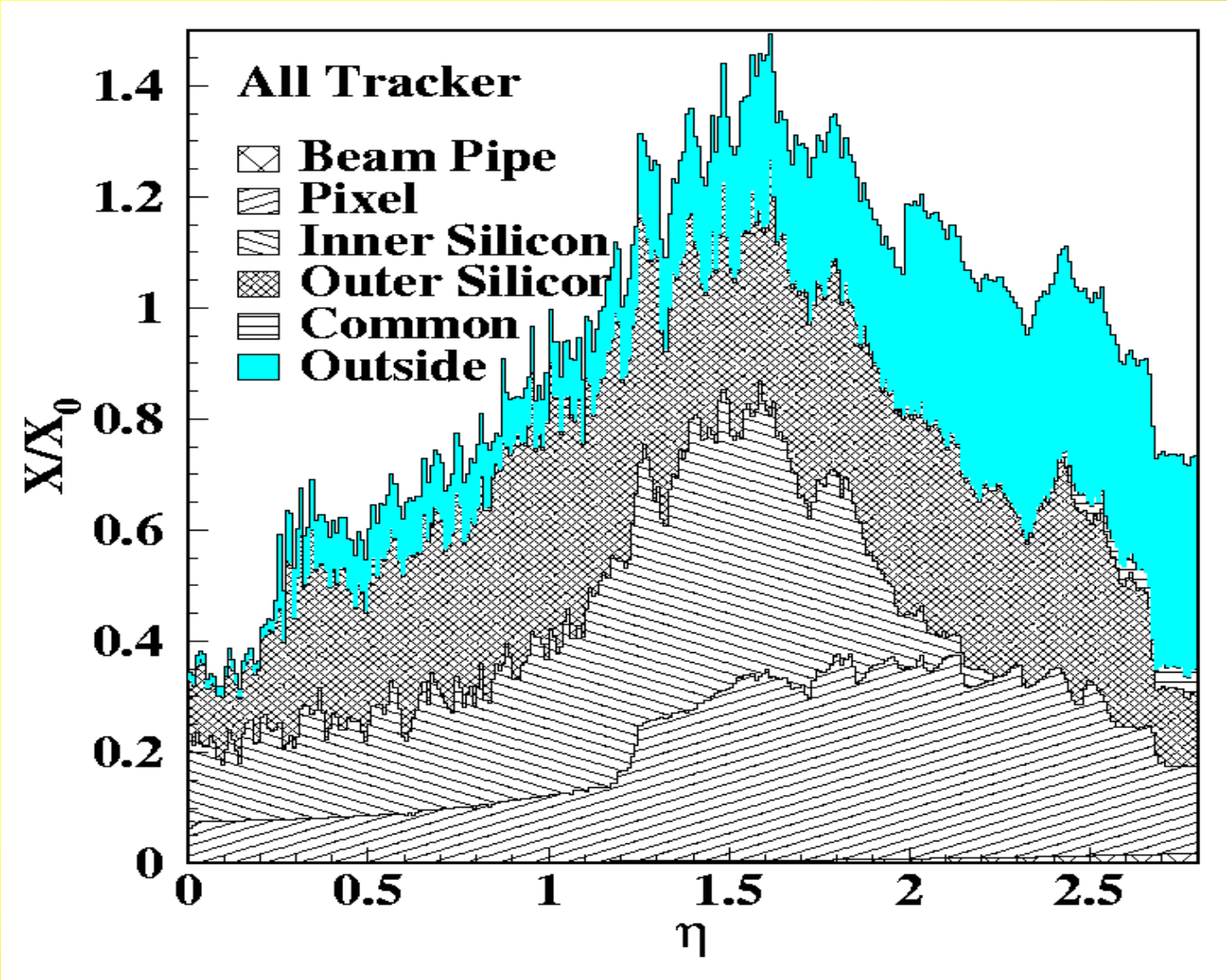
Teddy Todorov

What is special about CMS tracker

- Few measurements
  - ~13 measuring layers (3 pixel, 4 double-sided strips)
- High quality individual measurements
  - 10 to 45 u resolution
  - largest channel occupancy in high lumi pp below 5%
  - few hundred microns two track separation
- A lot of material per measurement layer
- LHC scale
  - zillions of channels, bondings, hits per crossing, ...
  - most hits *not* from tracks of physics interest (very low Pt, pile-up)

Total material in tracker volume

- A lot of material! Radiation lengths:



Material estimation from MC

$\Sigma\ \Delta s/x0*\sin\vartheta >$ vs. radius for $|\eta|<1$

Black:

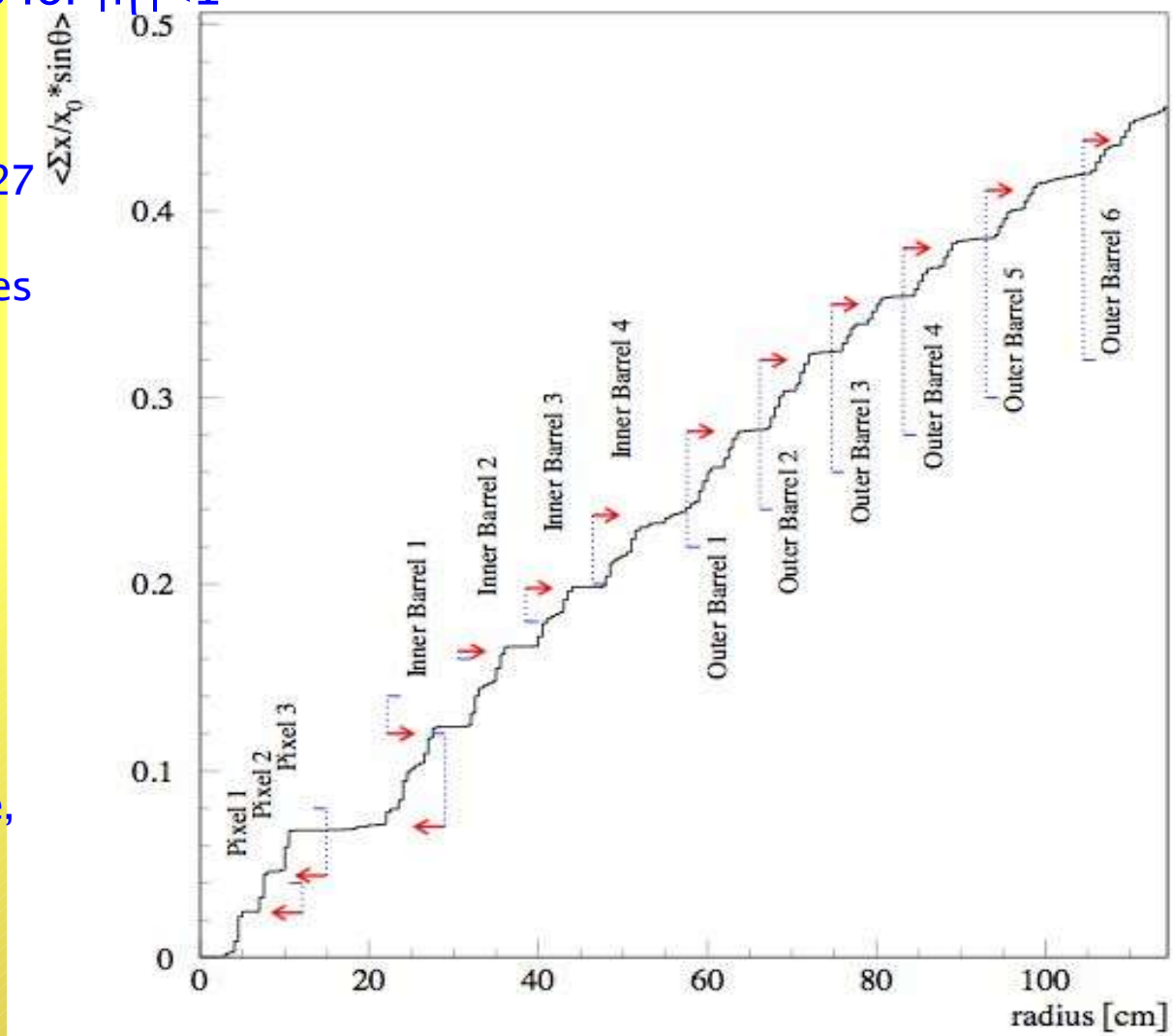
< s/x0*sin > from CMS127

Red:

estimate used for GSF studies
(fit to step sizes)

Blue:

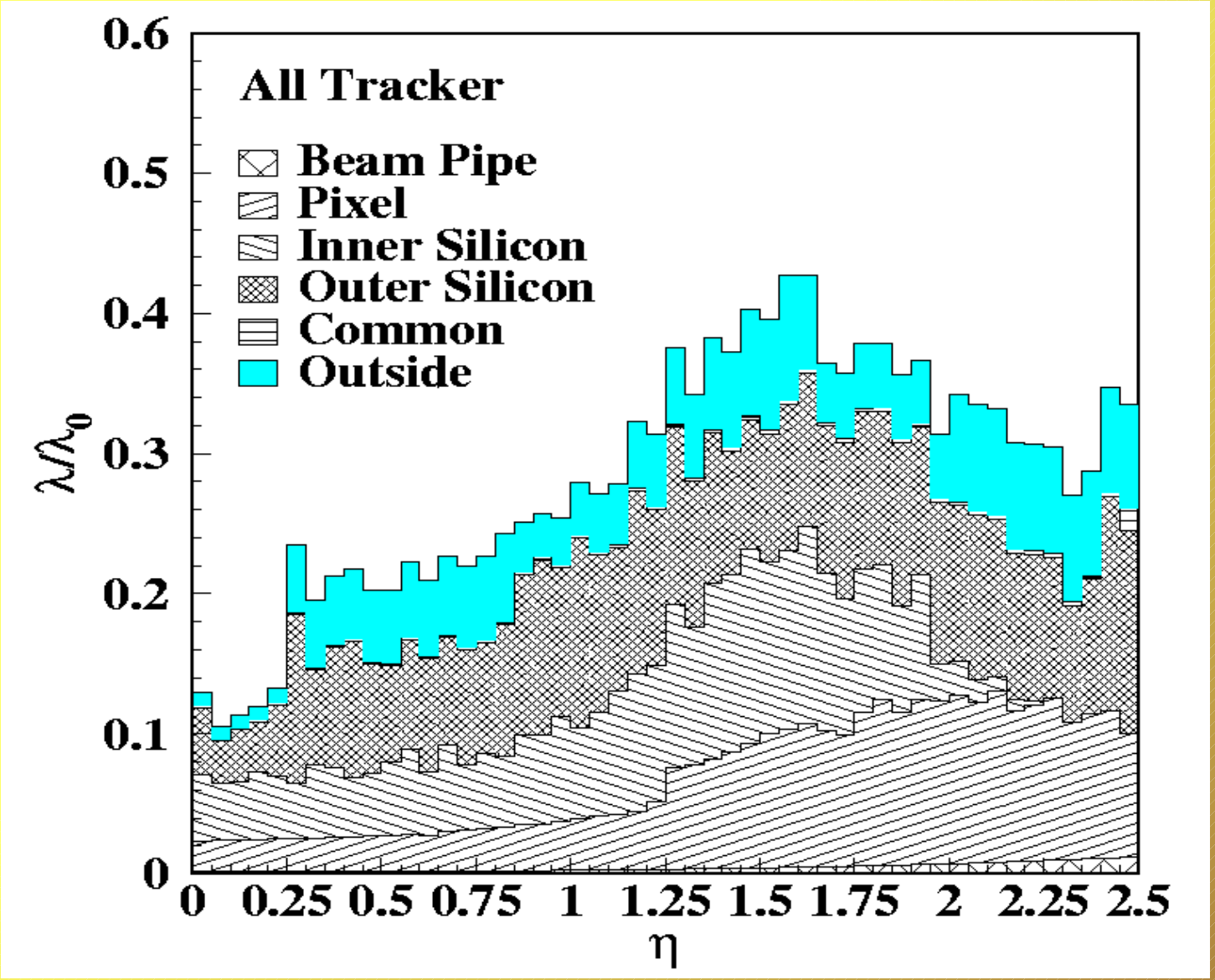
default values



Fit results depend on eta-range, vertex smearing, …!

Interaction lengths

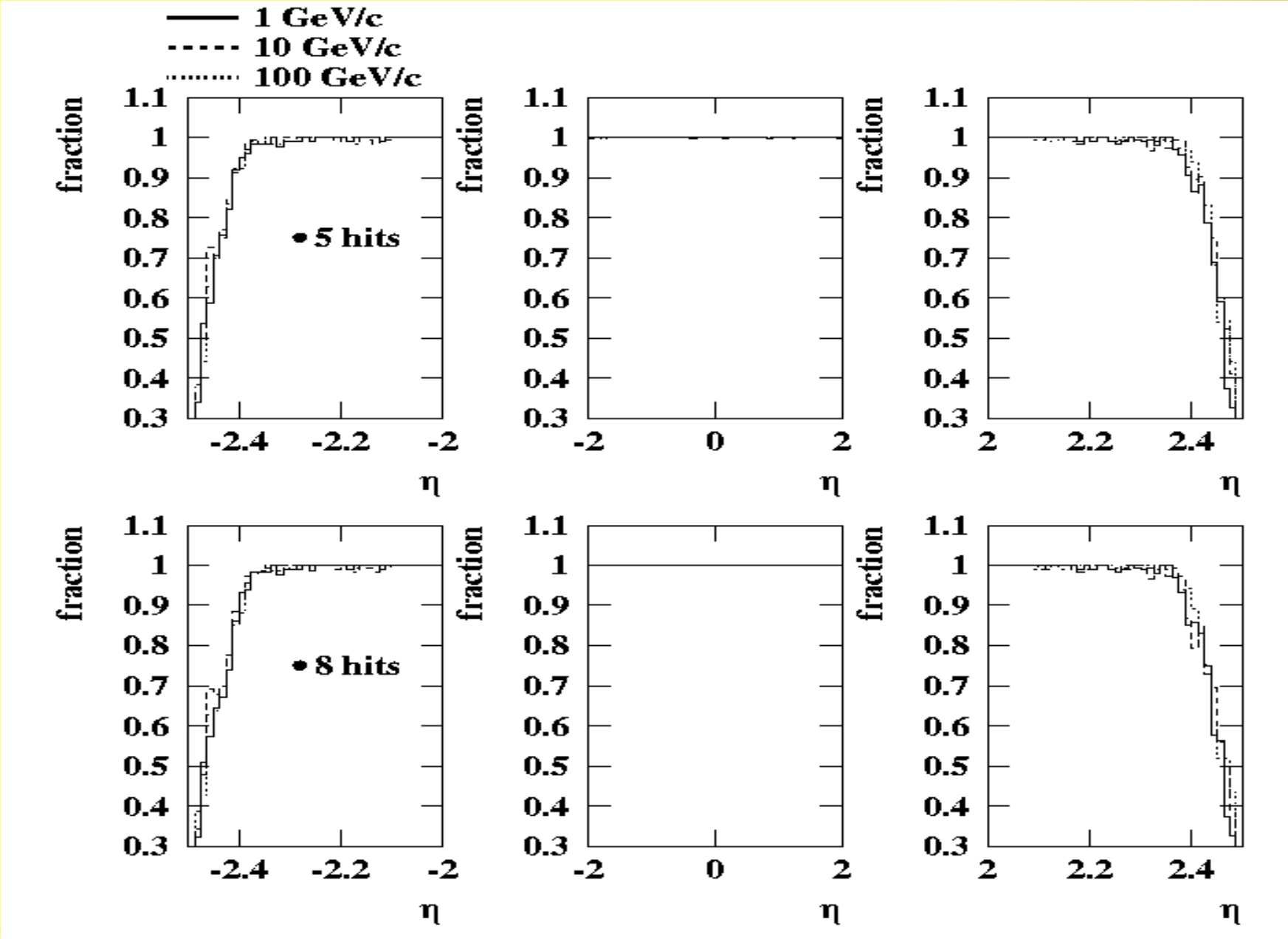
- More dangerous: kill tracks



Teddy Todorov

Probability to cross 5 or 8 layers: muons

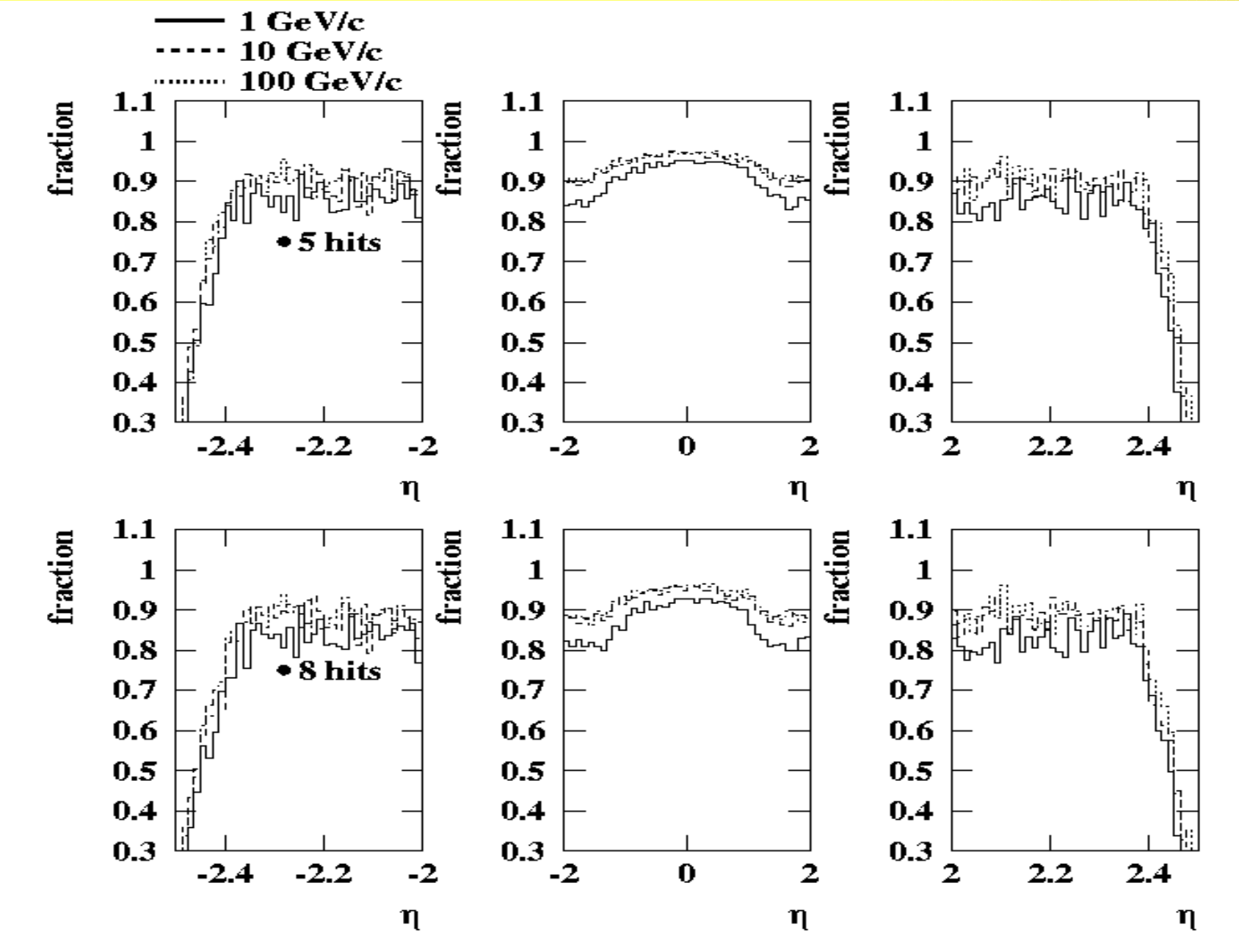
- Muons are mostly unaffected (ouf!)



Teddy Todorov

Probability to cross 5 or 8 layers: pions

- Pions suffer substential losses



Teddy Todorov

Impact of material on tracking strategies

- Impossible to ignore material effects during pattern recognition
  - Kalman filter or better required
- Makes no sense (for primary tracks) to track from outside in
  - up to 20% of tracks just don't reach the “outside”

Tracking problem decomposition

- Kalman filter inspired (but applies to a large family of algorithms)
  - seed generation
  - pattern recognition for individual seeds (trajectory building)
  - resolution of ambiguities (trajectory cleaning)
  - final track fit (trajectory smoothing)

Trajectory seeding

- A seed is a trajectory state
  - corresponding (usually) to a single track
  - constrained in all 5 parameters to allow meaningful search for compatible hits on tracker layers
    - usually contains at least one tracker hit
  - with all 5 parameters reasonably close to true values (linear regime)
- A seed can be internal to the tracker or external (muon system or calorimeter information)
- A region (like jet cone) is *not* a trajectory seed
  - but it can be used as input for seed generation (see regional reconstruction)

Internal seeding for primary tracks

- To constrain all 5 track parameters one needs e.g.
  - two RecHits with 2D information
  - interaction region and minimal Pt cut
- Pixel hits a natural choice
  - closest to the IP – least affected by material effects
  - very low occupancy
  - very low noise
  - redundancy when using any 2 out of 3 layers
  - very precise and unambiguous 2D measurements

Pixel seed generation

- Geometrical hit pairing based on a cylindrical interaction region and a minimal Pt cut
- The size (radius and length) of the interaction region is a free parameter:
  - e.g. 1-2 mm in radius for tracks from b jets, 50 u for prompt muons
  - 15 cm in z if primary vertex unknown, max(radius,primary vertex resolution) if known
- Very efficient implementation for hit pair search: O(100 ms/event) for high lumi pp collisions and “global” region
- Main problem: too many seeds: O(30k) at high lumi
  - only O(100) correspond to real tracks from trigger event

Pixel primary vertex reconstruction

- If the primary vertex position of the trigger event is known before track reconstruction, it can be used as input to the seed generation
  - large reduction in number of ghost seeds (factor ~100)
- PV easily known for muon and electron triggers
  - via regional mu/e reconstruction, see later
- In the absence of a suitable trigger, the primary vertex of the "trigger" event can be found with >99% efficiency for most physics channels
  - by using pixel triplets

Status of pixel PV reconstruction

- Mature
- Well optimized (O(100 ms) at high lumi)
- Tuning of the tagging of the trigger vertex for specific physics channels a long-term activity
- Pixel triplets (obviously) usable as high purity seeds
  - but efficiency is about 10% lower due to geometrical and readout inefficiencies of the pixel system
  - and thus not very useful, except for special purposes like Heavy Ion collisions

Teddy Todorov

Summary of seed generation

- Fully efficient
- (too) highly optimised
- Primary vertex information results in reasonable umber of seeds at all luminosities
- Depends on 3 pixel layers for optimal performance
  - also works reasonably with 2 pixel plus one silicon layer

Teddy Todorov

Trajectory building

- Straightforward logic:
  - starting from a partial trajectory of N hits (initially a seed)
    - find all *compatible* hits on the next directly reachable layer(s)
    - for each compatible hit consider an alternative trajectory of N+1 hits
    - do something to limit the total number of alternative trajectories
    - repeat the procedure for all N+1 hit trajectories
  - trajectories stop
    - when they don't have compatible hits on the next layer(s)
      - not immediately, they are allowed to miss a configurable number of hits
    - when the last tracker layer is reached

Invalid hits

- To make the pattern recognition fully unbiased, even if there are compatible hits on a layer the alternative of a missing hit is systematically considered
  - modeled as an “invalid” hit which does not have position information but simply records on which sensor and layer a hit is missing

Hit compatibility

- The measure of compatibility of a hit with a trajectory state is an abstract component, MeasurementEstimator
  - can be any criteria
  - for Kalman filter is Chi2

Limitations of the pattern recognition

- Given
  - a fully efficient search for compatible hits
  - a fully efficient layer navigation

the pattern recognition can have algorithmic efficiency arbitrarily close to 100%

  - limited at this stage only by
    - the “Chi2 cut”
    - the number of alternative trajectories retained
    - the number of missing hits tolerated
- There is no bias
  - does not stop when a good track is found
  - hits are not locked
- This may take arbitrarily large CPU time, of course :-)

Example of number of candidates

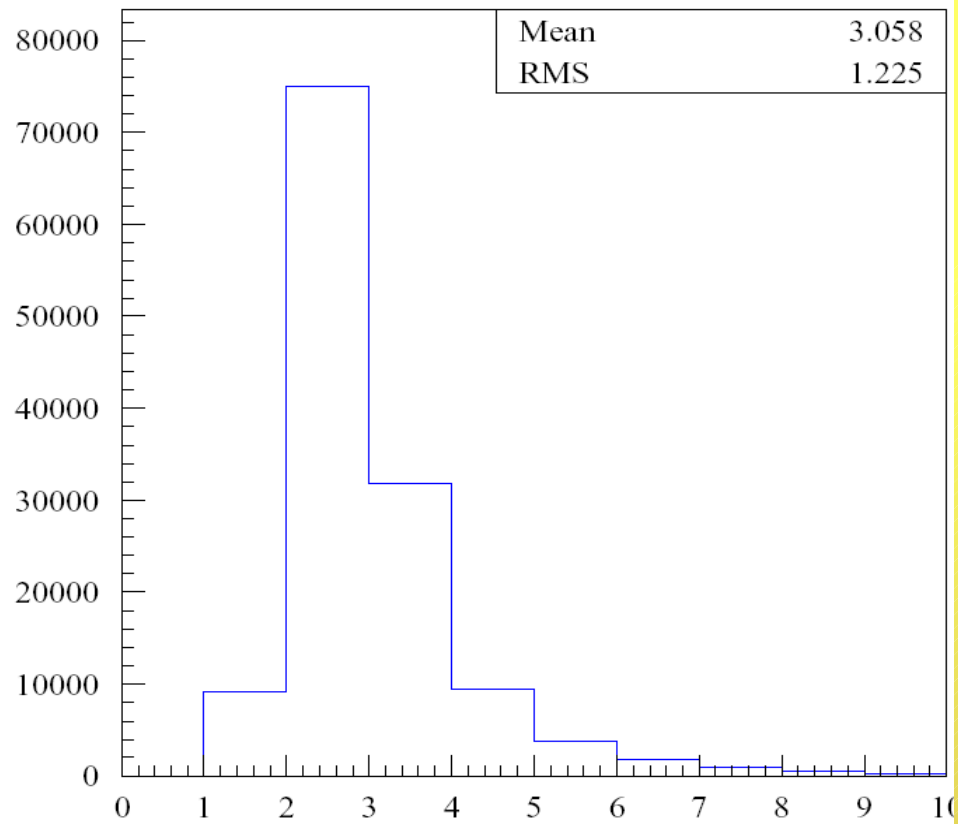


Figure 14: *Number of compatible hits found on TIB layer 1 for each trajectory candidate when leaving Barrel Pixel layer 3 for 100 GeV b jets without pile up.*

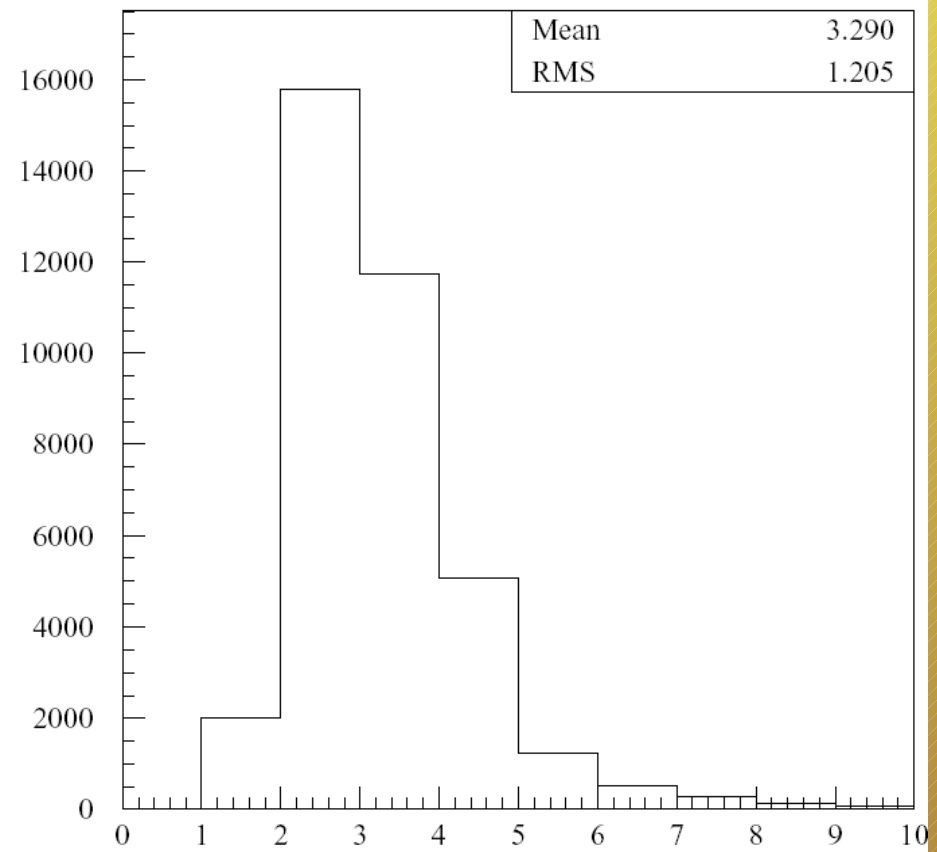


Figure 21: *Number of compatible hits for each trajectory candidate when leaving the Forward Pixel disk 2 (100 GeV b jets without pile up).*

Limiting the combinatorial growth

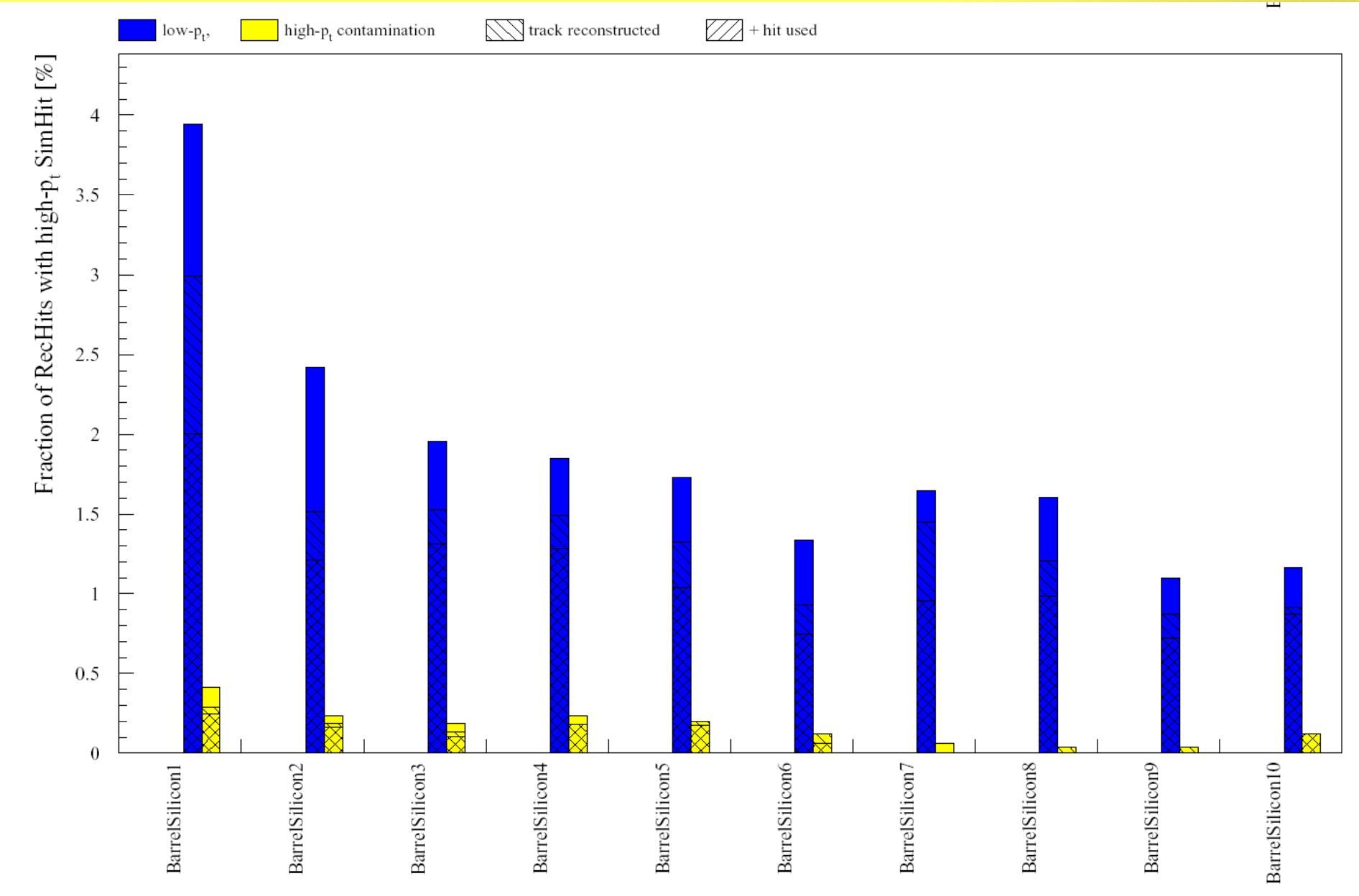
- The first thing to do is to use a restrictive MeasurementEstimator
  - However, real RecHit residuals don't have Gaussian shapes even for single track RecHits, and in dense jets RecHits can be polluted by other tracks
  - A generous Chi2 cut (default at 30) is needed for “full algorithmic efficiency” of collecting compatible hits
    - “full efficiency” means “order of 0.1% inefficiency”

      Controlled via the MeasurementEstimator you give to the constructor, or

      via .orcarc: CombinatorialTrajectoryBuilder:chiSquareCut

RecHit pollution

For tracks in 100 GeV P_t b jets with pile up



Teddy Todorov

Limiting the combinatorial growth

- The best way we found to limit the growth found so far (idea came from HERAB HLT) is
  - grow all candidate trajectories in parallel, one layer at a time
  - At each layer sort the candidate trajectories by “quality”
    - Chi2 + lost hit penalty (all candidates have the same number of hits at this stage)
  - Before leaving a layer, limit the number of candidate trajectories to MaxCand (default is 5)

Reality vs. perfection

- It turns out that with a modest number of alternative trajectories and a reasonably generous Chi2 cut we can be within O(0.1%) from asymptotic efficiency for muons (and surviving pions), with acceptable CPU time consumption
  - the tuning of efficiency/CPU is obviously different for HLT tracking
- The CMS tracker is a very clean detector, even at high LHC luminosity
  - to the extent to which we know it's behavior in simulation, of course
  - real data will likely be less clean :-)

Trajectory building decomposition

- The pattern recognition is done in terms of layers:
  - detector layer is responsible for finding of compatible hits
  - the layer is also responsible for finding the next immediately reachable layer(s) along a track candidate
- The CMS tracker is not homogeneous
  - 6 different layer types, with completely unrelated geometrical structure
- Natural polymorphic behavior
  - uniform layer interface: no barrel/endcap or pixel/strip differences as far as tracking is concerned
  - each layer type hand-optimized according to it's geometrical structure, symmetries, etc.

More decomposition

- The act of incorporating the information from a measurement (hit) into a predicted trajectory state is called *updating* in CMS parlance (Kalman inspired)
  - The Updator is an abstract component, the KalmanUpdator is just one possible specialization
- The propagation (or transportation) of a trajectory state to a different surface is done by a *Propagator* component
  - atomic action: parameters, error matrix, material effects all in one invocation
- The compatibility of a hit with a trajectory state is defined by a *MeasurementEstimator* component

Ambiguity resolution

- A single seed typically results in several candidate trajectories
  - sometimes there are more than one “true” tracks
- The ambiguity resolution is very simple: trajectory groups with more than 50% of shared hits are reduced to one (best) trajectory

Final track fit

- Straightforward sequence of propagations and updates with the hits found during pattern recognition
- Forward + backward fit with combination of results (Smoothing)
- Any bias introduced by a constrained seed (beam spot, etc) is removed at this stage
  - the fit is redone from scratch

Regional track reconstruction

- In the CMS DAQ there is ***no*** restriction on raw data access at HLT level: the entire event is available at 100 kHz rate
  - Regionality is purely logical, not constrained by hardware/DAQ
- Since track reconstruction proceeds one seed at at time all stages after seed generation are “regional”
- A tracking region is a volume in 5D space
  - more or less abstract component

Regional seed geeration

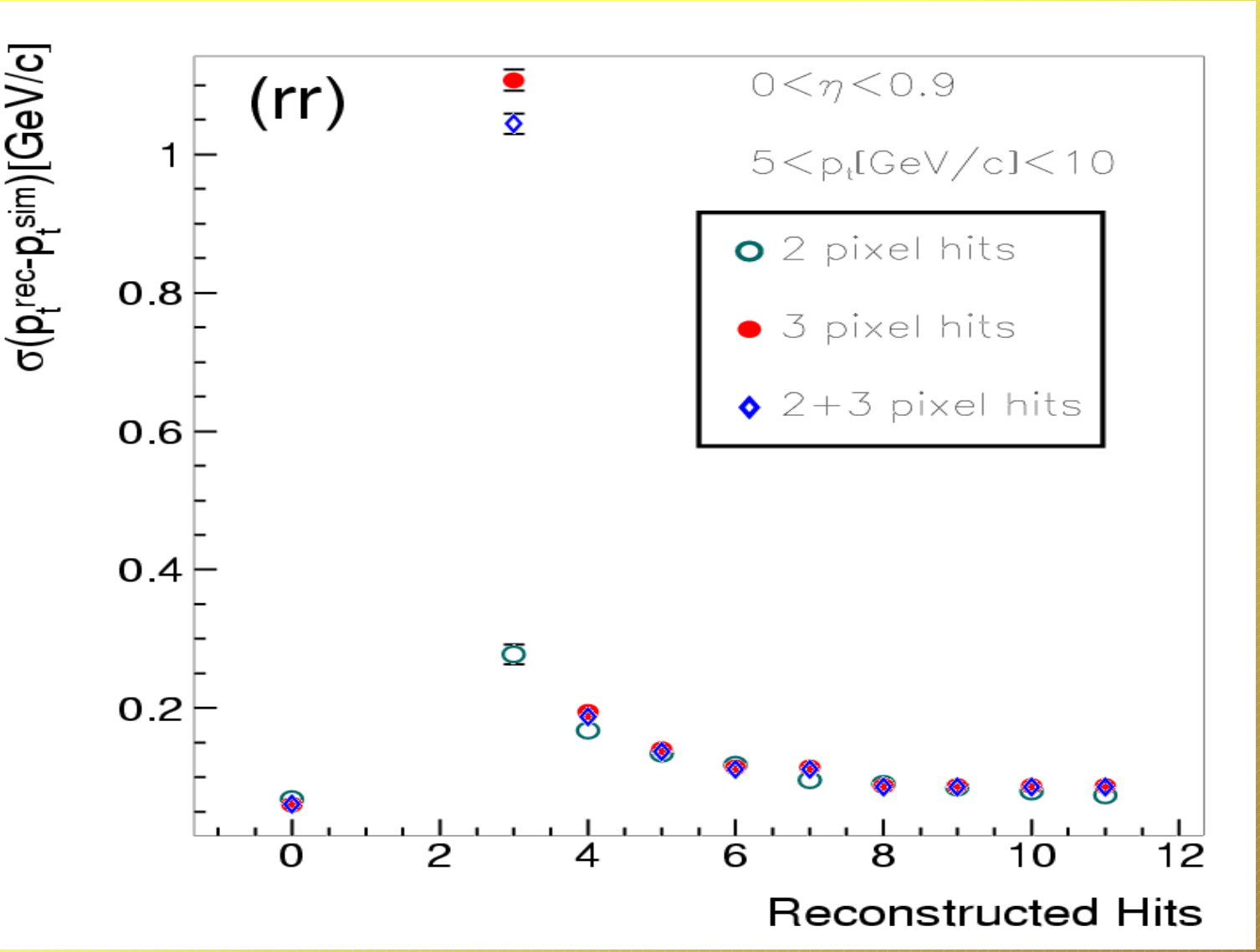
- Seed generation is *always* regional:
  - interaction region has final size
  - there is always a momentum (minimal Pt) cut
  - “Global” region means the directions of tracks are not constrained
  - “Regional” reconstruction means additional constraints in eta/phi
- The region is fully used in the seed generation algorithm
  - CPU speed gains proportional to region volume reduction

Regional trajectory building

- in addition to the position an direction constraints from the region (used in the seed generation), there is a momentum constraint which cannot be enforced before a sufficient number of hits are collected
  - after each “update” the momentum is checked against the cut

User stopping conditions

Sufficient accuracy may be reached before the end of the tracker:



Teddy Todorov

Conclusions

- The baseline track reconstruction in CMS is a combinatorial Kalman filter algorithm
- A large number of configurable options allows for smooth transition from optimal performance (efficiency/precision) to high speed in a controlled way
  - HLT use demonstrated for muon, tau, and B tagging of jets
  - Even used in heavy ion
- The Kalman filter is a benchmark for more advanced algorithms

Outstanding issues

- Alignment algorithms
  - the *mis*alignment framework is quite advanced:
    - misalignment can be specified at every level of the detector hierarchy (from individual sensor to the entire barrel)
    - uncertainties on alignment (detector positions) can be specified
    - the track fit automatically takes misalignment and uncertainties into account
    - coherent system, also used by muon
  - the alignment work has yet to start
    - some very promising ideas
    - some implementations at test beam scale

Outstanding issues (2)

- Dead channel treatment in reconstruction
  - hit position/error modifications
  - track reconstruction “expectation lowering”
- Systematic treatment of calibrations
  - at all levels
  - miscalibrations on Monte Carlo

Organizational aspects

- Single track reconstruction framework
  - no “one man – one algorithm” problem
  - no duplication of functionality
    - e.g. *all* algorithms use the same set of propagators
- Same framework in Tracker and in Muon
  - transparent global muon tracking
  - quite general framework