A Level-1 Track Trigger for CMS with double stack detectors and long barrel approach

---

Emmanuele Salvati
Cornell University
on behalf of the CMS collaboration

WIT May 3, 2012

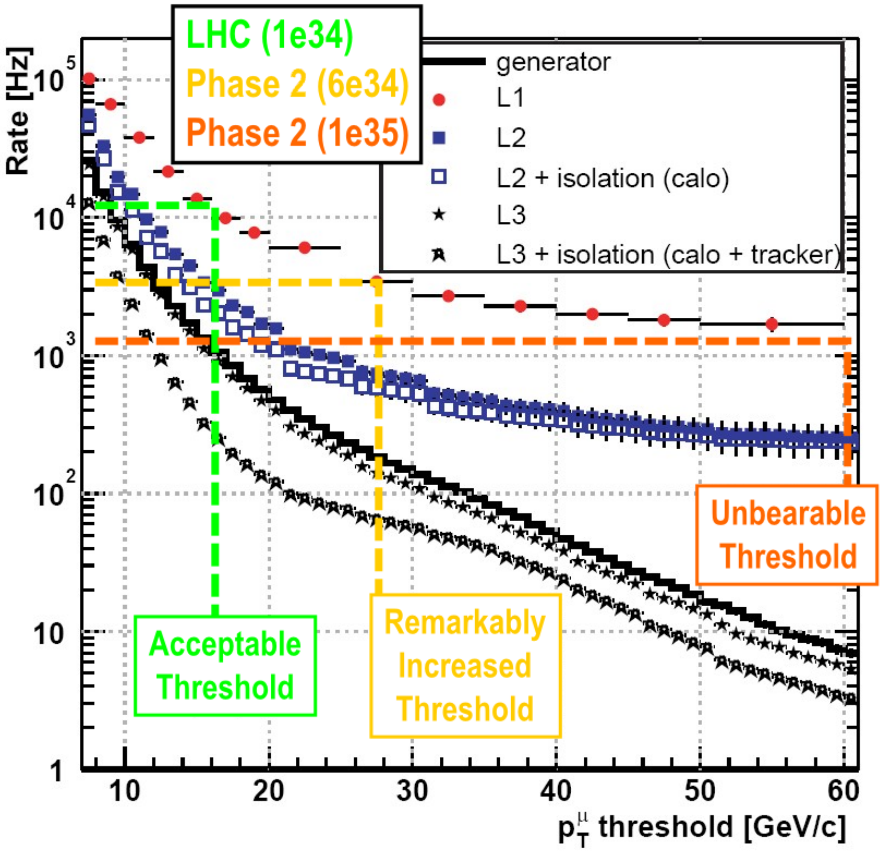
Outline

- L1 tracking trigger concept
- Long barrel geometry
- Definition of L1 primitives
- Hit rates
- Track reconstruction at L1

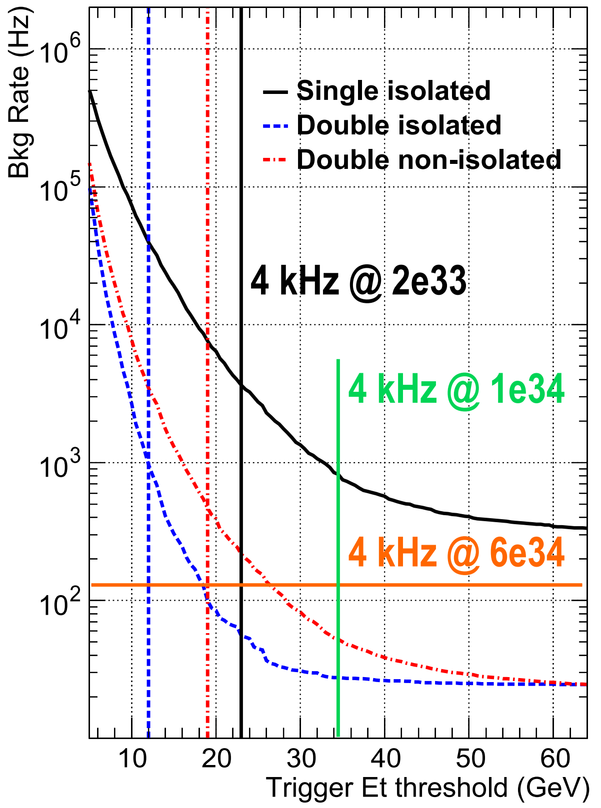
Triggering at LHC phase 2

- Tracker input to L1 trigger is necessary
  - combined mu, e and jet triggers would exceed 100kHz at high luminosity and pile-up
  - increasing thresholds would affect physics performance
  - including tracks pT measurement from the tracker reduces significantly the rates

Estimate of mu triggers at high luminosity

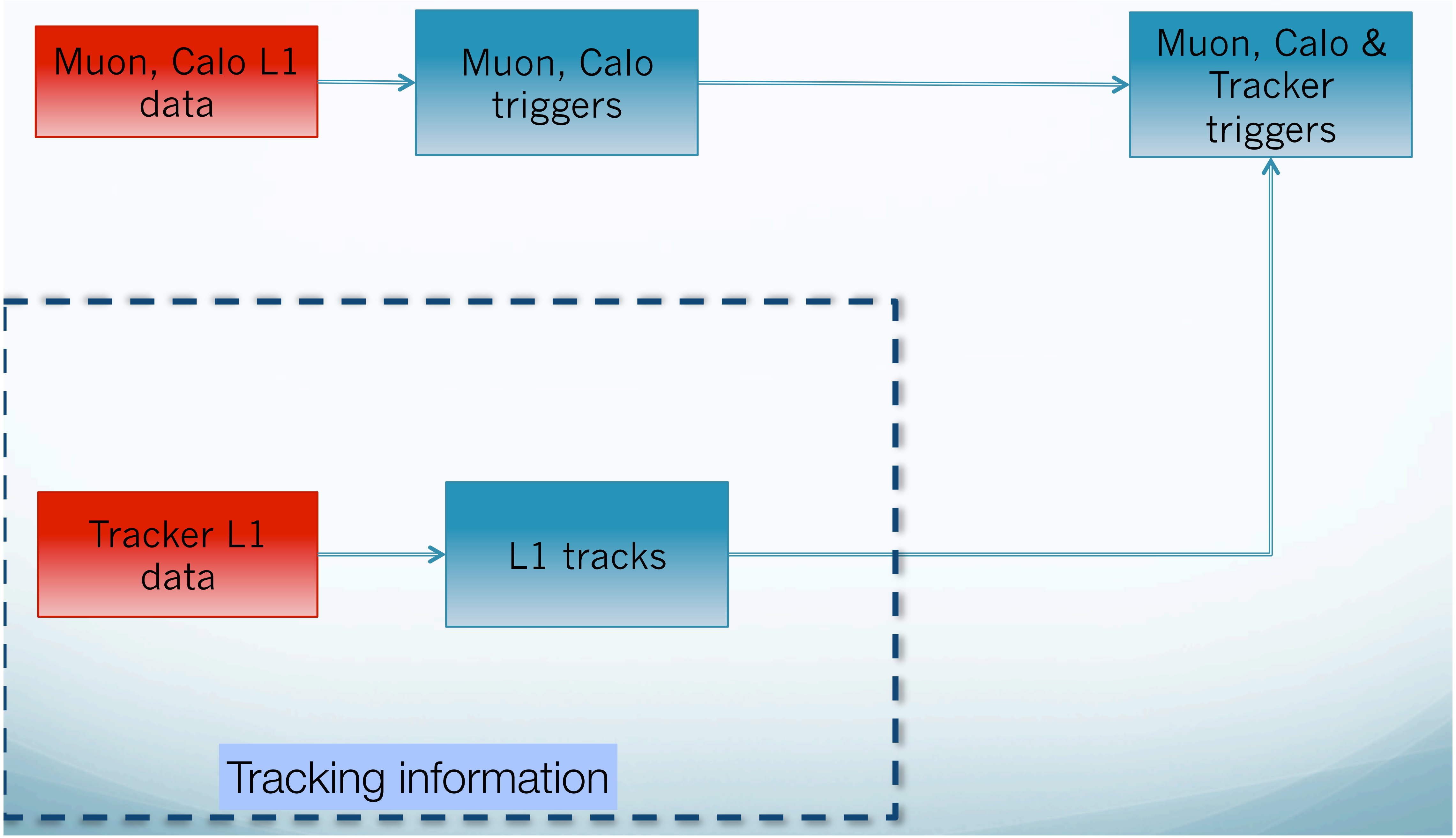


Estimate of calo triggers at high luminosity



out of plot range

Triggering at LHC phase 2



Muon, Calo L1 data

Muon, Calo triggers

Muon, Calo & Tracker triggers

Tracker L1 data

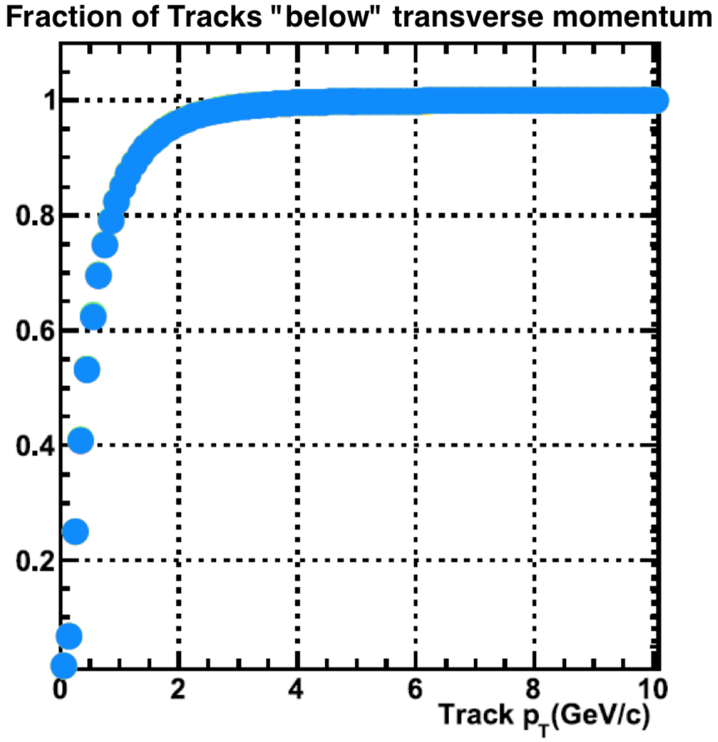
L1 tracks

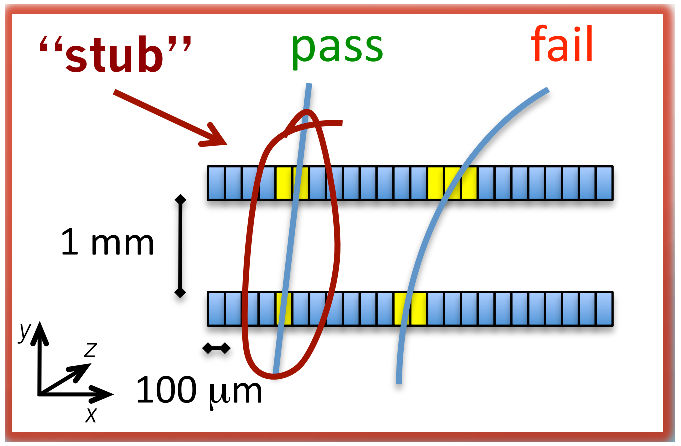
Tracking information

Track trigger concept

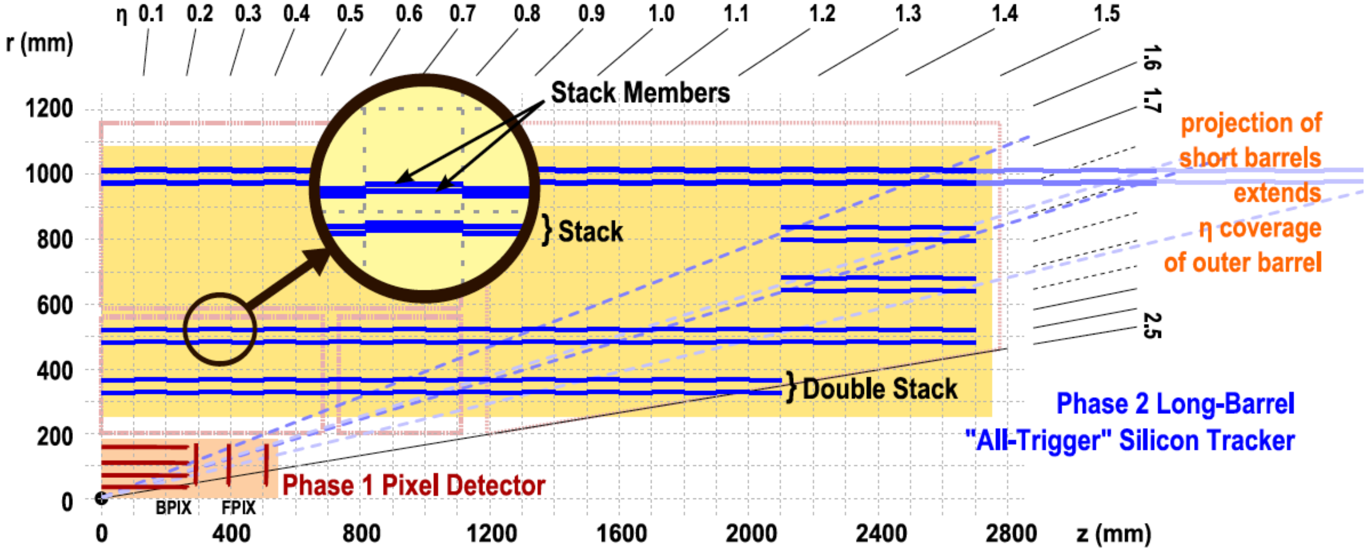
- **Objectives**
  - Reconstruct tracks with p_T > [2 - 2.5] GeV/c
  - Identify the origin along the beam axis with ~1mm precision
- Silicon modules to provide both Level-1 data (40MHz) and read-out data (~100kHz)
- Require local rejection of low p_T tracks for L1 data
  - very large data reduction
- Very fine z resolution due to pixels allows to remove large fraction of combinatorics background
- "p_T modules" for p_T discrimination
  - hit correlation in two sensors very closely spaced apart
  - exploit large CMS magnetic field
  - ~100µm resolution on lateral displacement is needed
- Define L1 stubs
  - minimum p_T-threshold for accept-reject stub
  - used as basic components of L1 track

~95% of tracks have pT<2.5 GeV/c





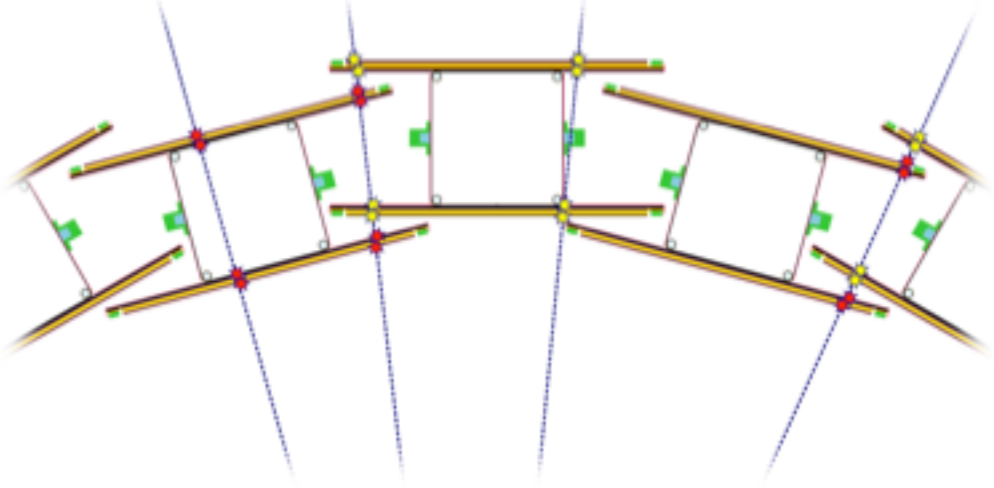
Long barrel geometry



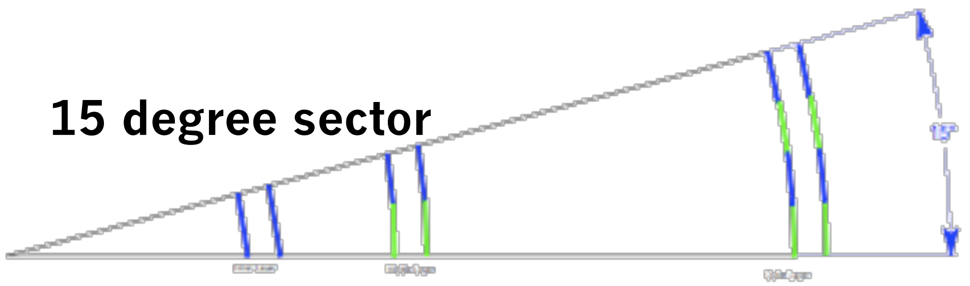
Outer tracker completely rebuilt with p_T modules

Current configuration

- 6 long barrels (= 3 Super Layers)
- Stack composed of two pixel layers ~1mm apart
- Double stack composed of two stacks ~4cm apart
- Pixel size: ~100µm x ~1mm (ϕ x z)
- Very challenging design
  - large number of pixels, high power consumption...
- Hermetic azimuthal coverage to keep data flow local within a ladder



Self-contained ϕ sectors.
Each sector needs to be combined with the two neighbouring sectors (left and right) to “contain” ~2.5 GeV tracks.



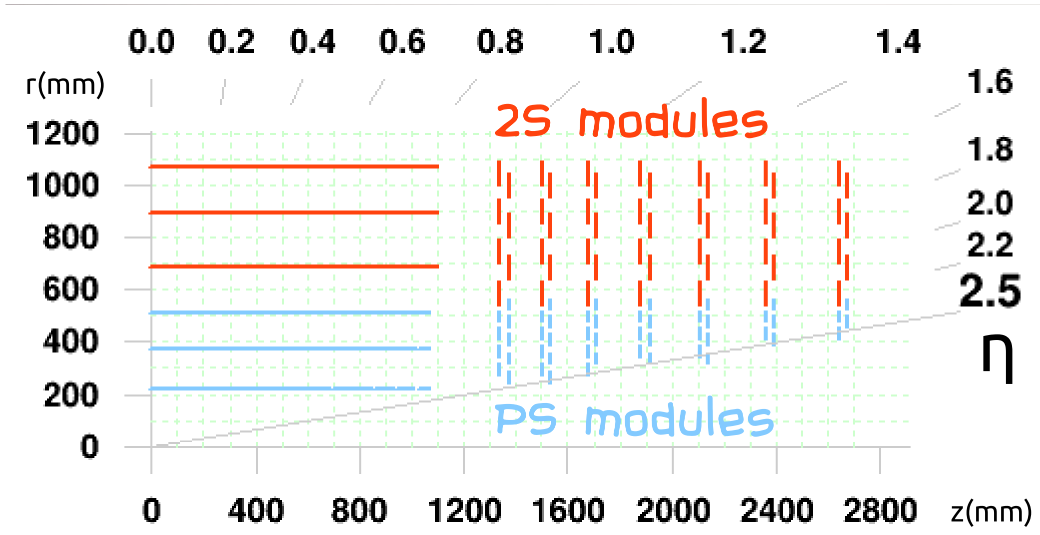
Features and limitations of the simulation

- **The track trigger code is flexible**
  - Implementation details of each object (i.e. cluster, stub, etc.) invisible to higher level objects
  - There are already several available algorithms for primitives
  - Physics: local minimum p_T threshold is configurable
  - Pixel size and stack separation are customizable
  - In principle it is possible to try different layouts (es. swap SL2 ↔ SL3)

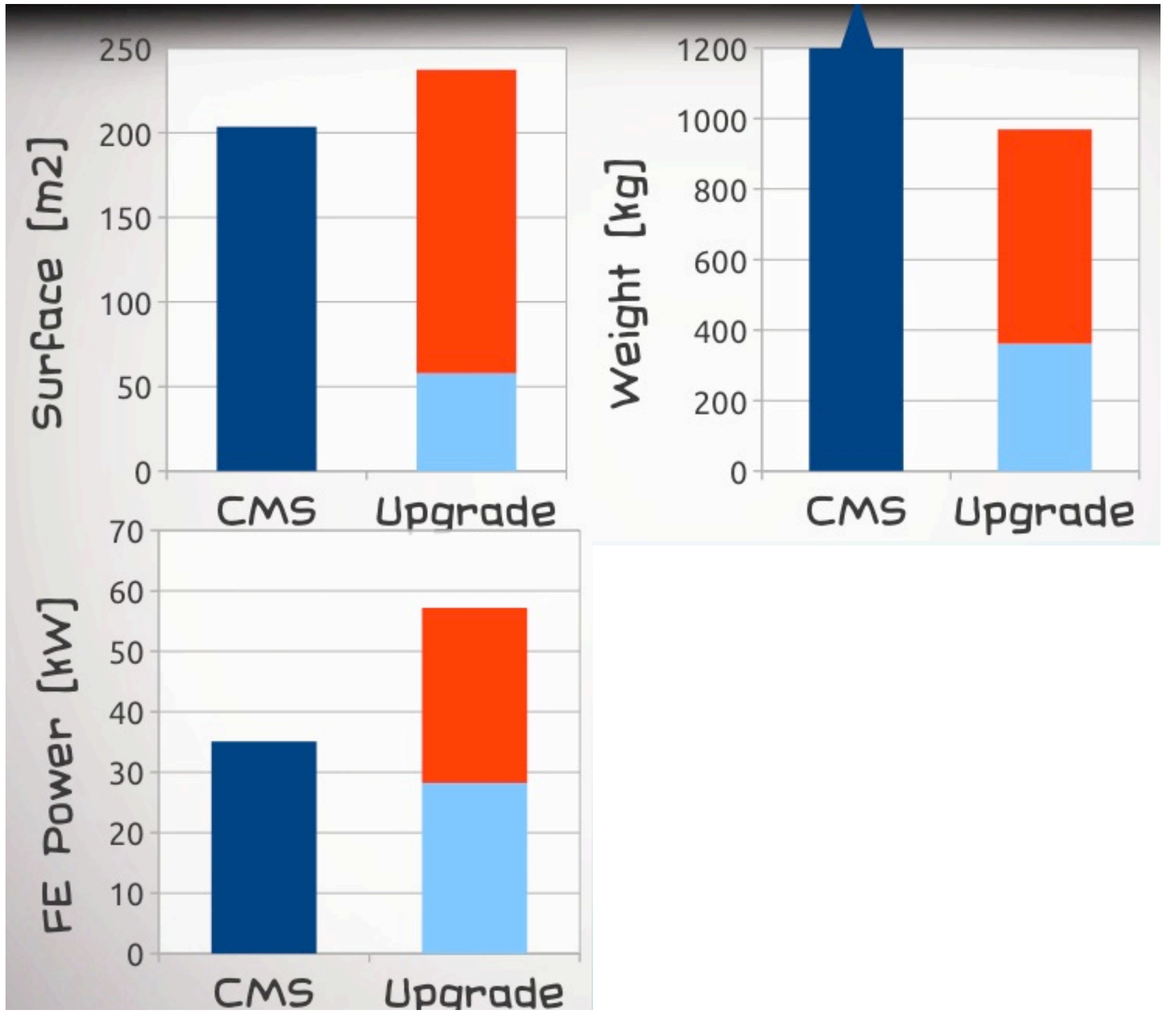
- **But track trigger code is still hard wired on the long barrel geometry**
  - Ongoing work to remove the dependency on the particular layout
  - Only pixel-pixel modules configuration is currently available
  - ...Full automatization for any possible geometry not really feasible

Tk layout tool to design geometries

- Useful and easy tool to design new layouts



- It provides useful estimates



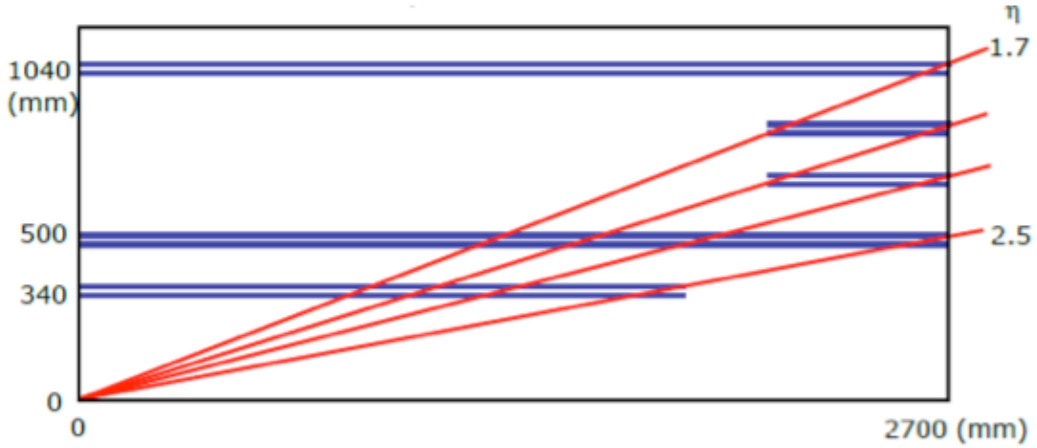
- It provides XML files to be included in the CMS software

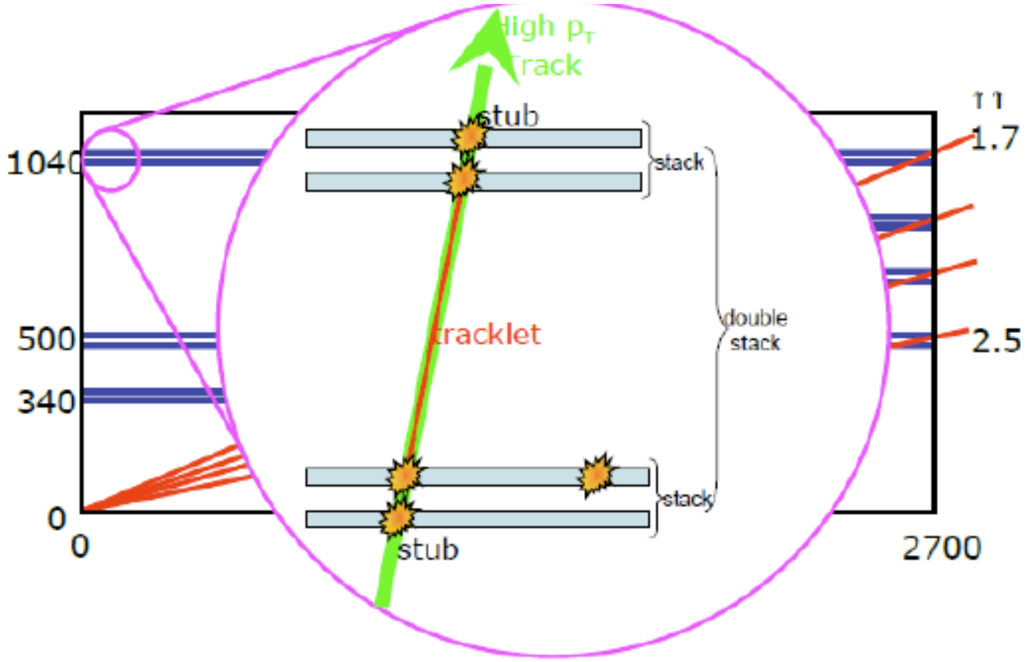
- **CAVEAT**: including those files into the CMS software is quite difficult
  - Fixes in this automation process are needed
  - We are working to redesign the long barrel
  - It requires the skills of an expert in geometries

Sequential scheme for pattern recognition

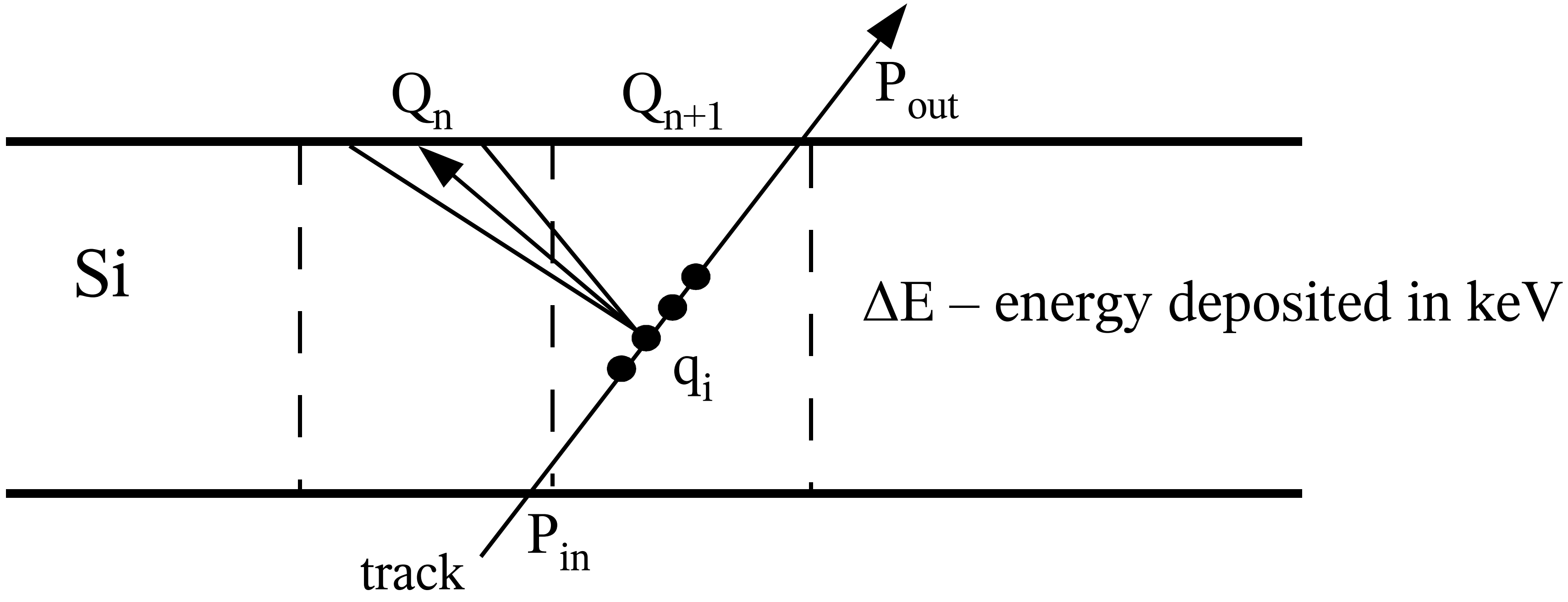
- **p_T sensor:** collect clusters of hits
  - in the simulation: generate sim hits for a given number of pile-up events
  - use pixel digitizer for ~realistic hits
  - clustering algorithm to remove combinatorics
- **p_T module:** two sensors ~1mm apart (layer)
  - Pair of clusters to form a stub
  - pT>2 GeV/c requirement to reduce rate
- **Super Layer:** two p_T modules ~4cm apart
  - Pair of stubs to form a tracklet
- **Current L1-track algorithm:** combine one tracklet with stubs in other layers
  - <u>L1-track algorithms still under development</u>

Each blue line is a p_T module (pair of sensors)





Pixel digitizer



(a) The track segment is split into many parts (about 100) each with charge q_i which is fluctuated according to the G4 dedx fluctuation formula.

(b) Each point charge is drifted to the detector surface under the influence of the B-field. (2nd order Lorentz force used).

(c) The point charge is diffused with a Gaussian smearing.

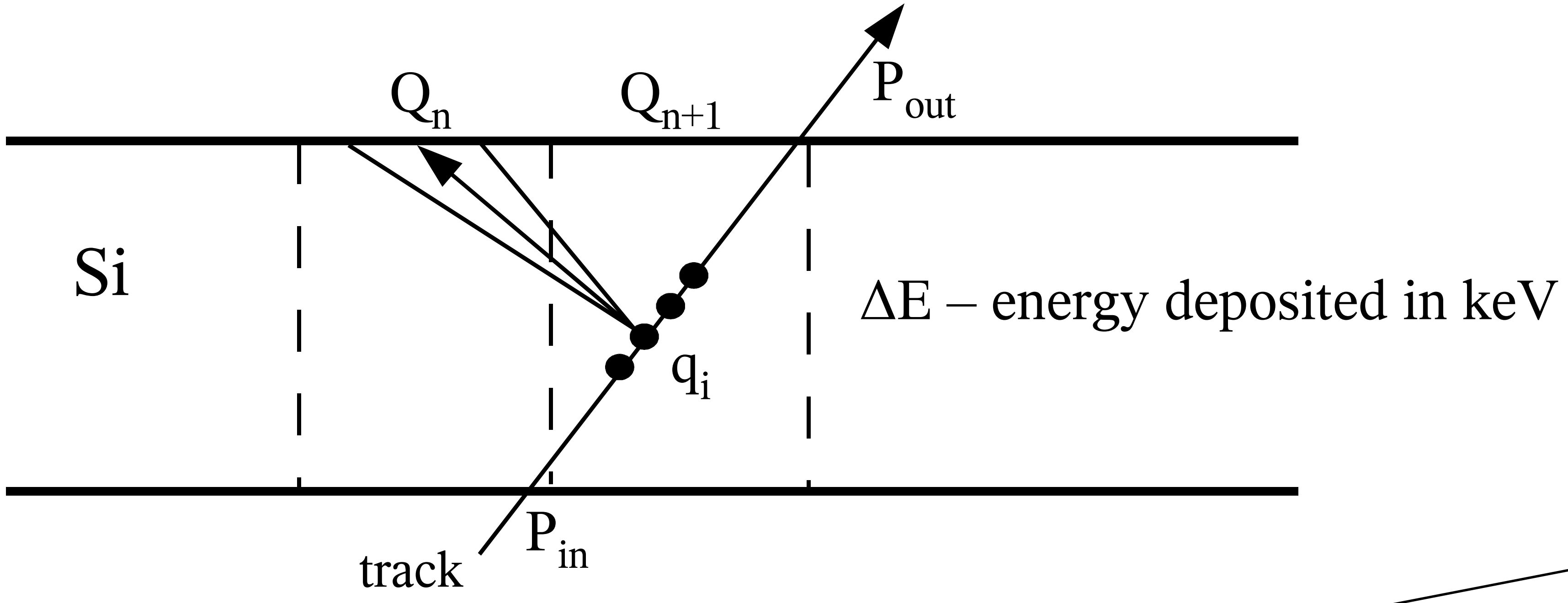
(d) All charges within a single pixel limit are collected to give the pixel charge Q_n.

(e) Noise is added. There are two types of noise: detector noise & readout noise.
The 1st one determines which pixel is above threshold and is read out.
The 2nd one determines the noise contributions to the signal at the ADC input.

(f) A threshold is applied and the charge is converted to ADC counts (integer).

(g) Inefficiencies and miss-calibration (ATANH formula) are applied.

Pixel digitizer



(a) The track segment is split into many parts (about 100) eac

which is fluctuated according to the G4

(b) Each point charge is dri

(2^{nd} orde

noise.

and is read out.

butions to the signal at the ADC input.

the charge is converted to ADC counts (integer).

s and miss-calibration (ATANH formula) are applied.

Digitizer concept is simple and reusable

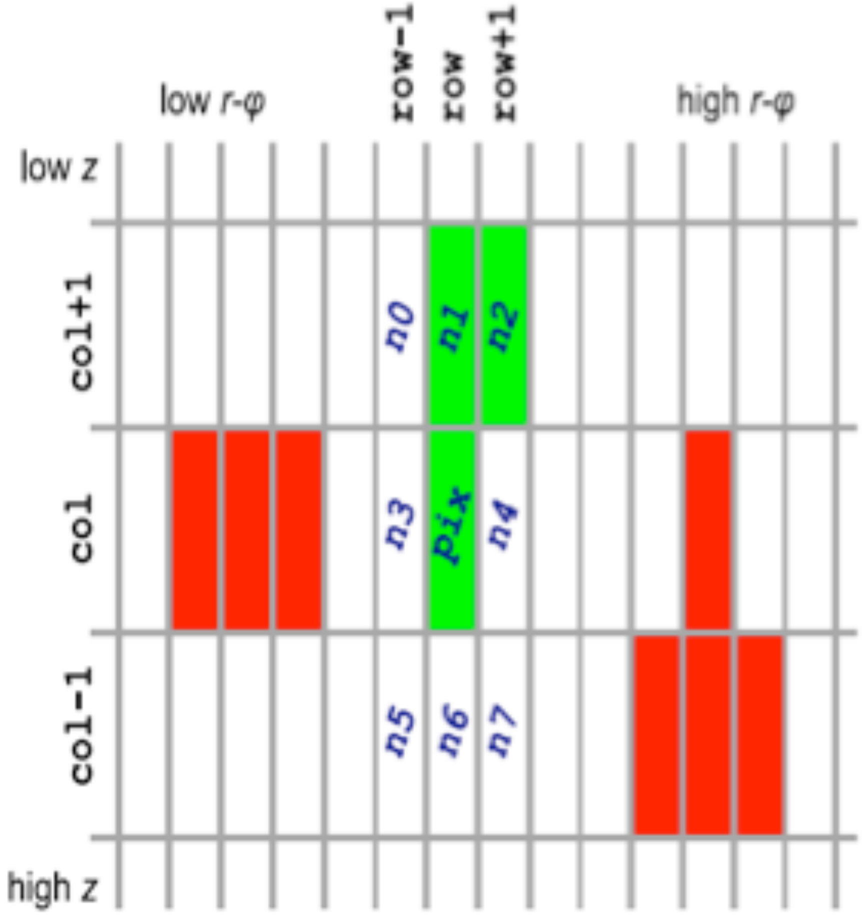
It easily accommodates larger pixel sizes

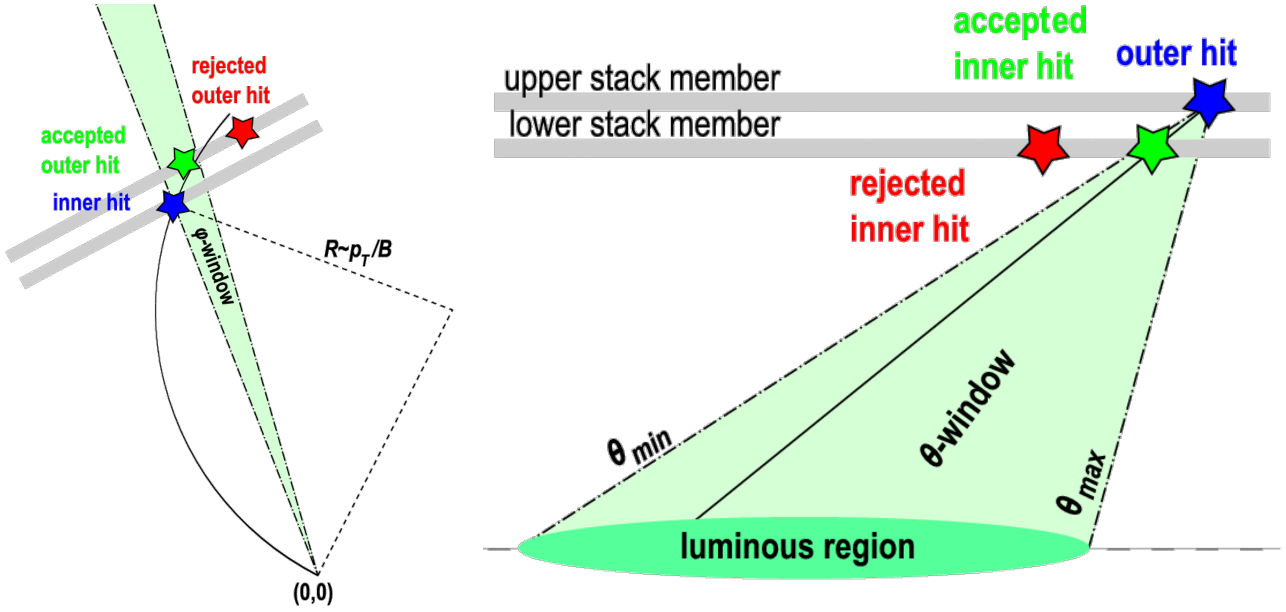
Physics input is fixed but customizable

Optimizing the digitizer is not high priority for the moment

Cluster and stub formation

- Hits need to be clustered, to reduce combinatoric bkg (2-3x)
- A few algorithms are available
- Currently using 2D algorithm





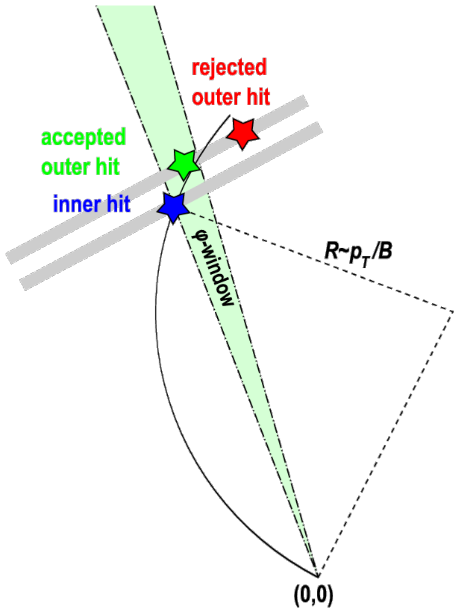
Clusters of hits are input information for production of track stubs

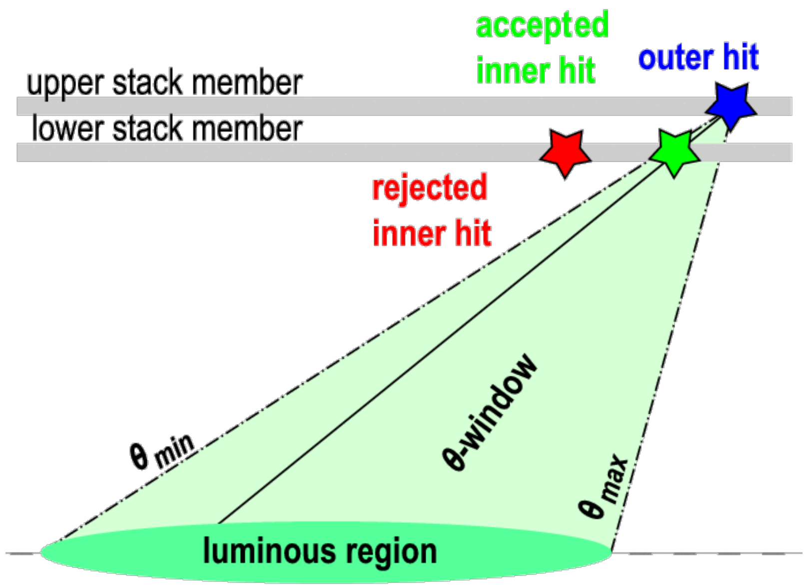
- **LUT's will depend on final tracker design**
- **use of global coordinates and trigonometry to open p_T-dependent matching windows**
  - **φ-window from p_T threshold**
  - **track must point back to luminous region**

Configurable p_T threshold in the code

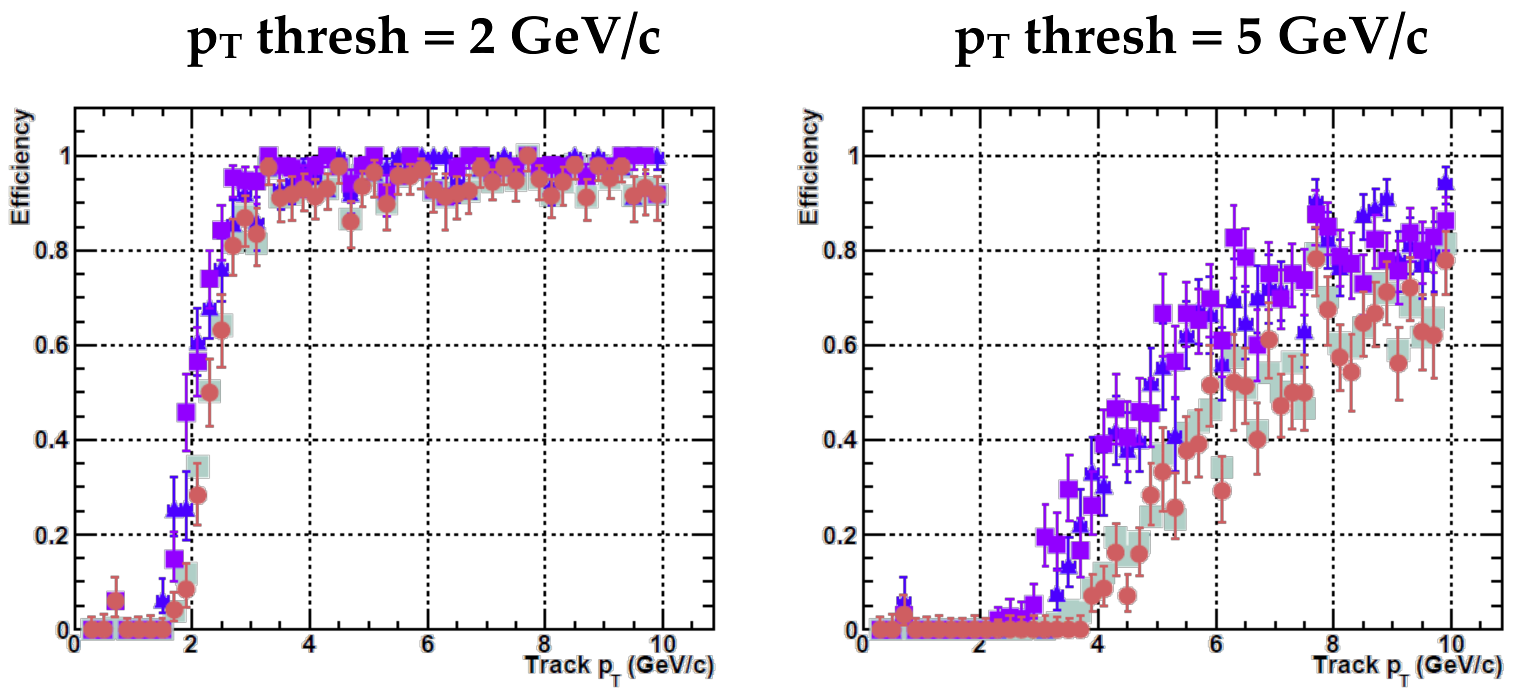
Stubs production efficiency

- Measured on single muon events
- Sharp production threshold for tight p_T requirement in the stub algorithm
  - **but much smoother for the higher (5 GeV/c) threshold**





- **Layer 1**
- **Layer 2**
- **Product of efficiencies in the two layers**
- **Product of efficiencies in the two layers requiring stubs in the same ladder**

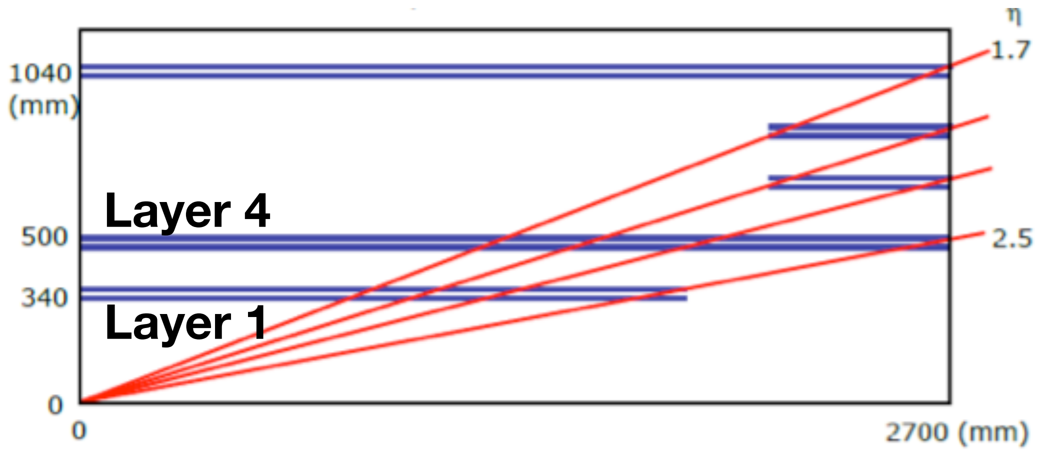


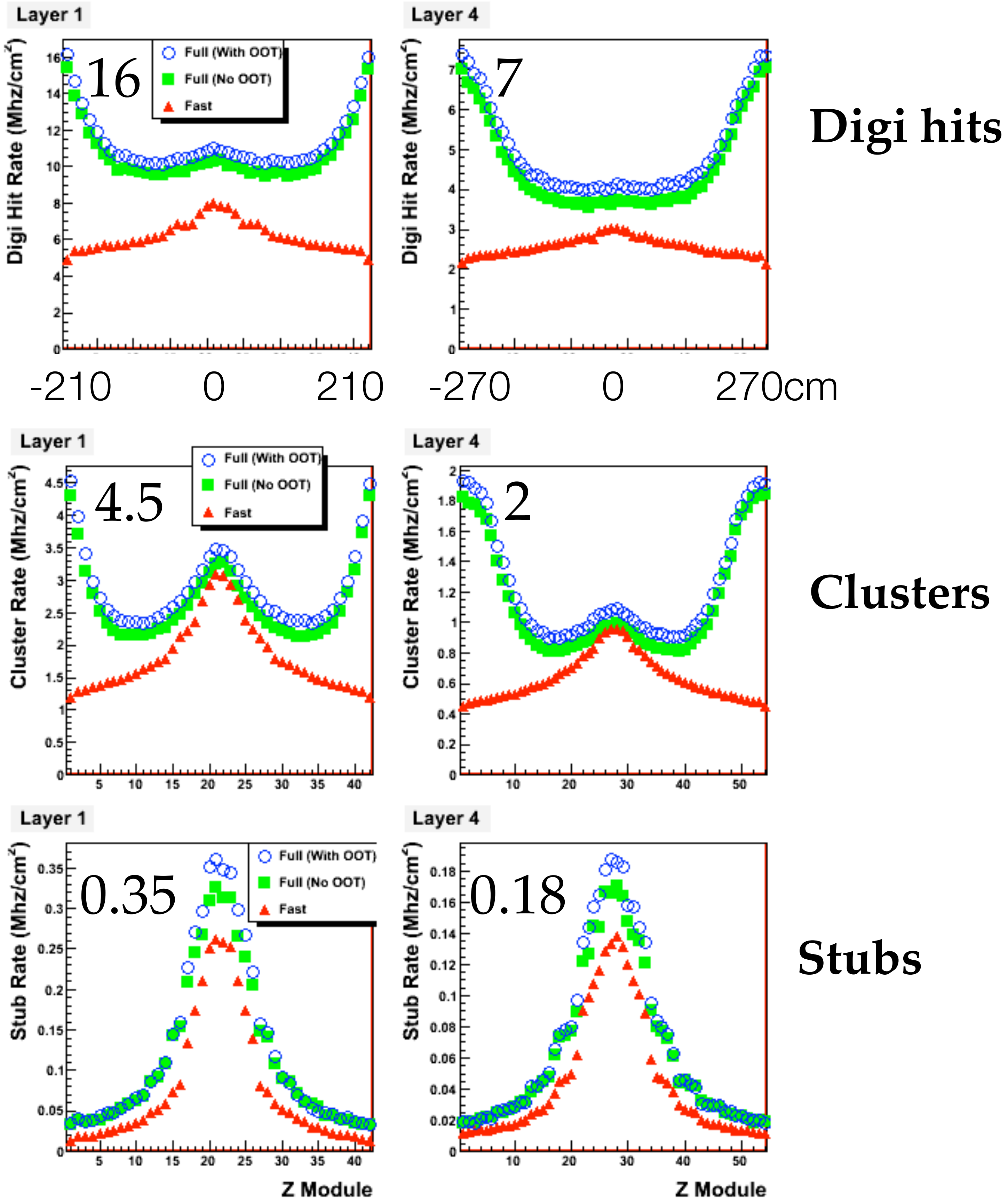
Stub production efficiencies on layers 1 and 2

Average rate estimates per module

- Each module has area ~100cm^2
- Rates averaged over phi
- Expressed in MHz/cm^2
- Calculated for 200 pile-up events per bunch crossing with 50ns bunch spacing
  - corresponds to expected Phase 2 luminosity ~ 5.5 10^{34} cm^{-2} s^{-1}
- Obtained with both full and fast simulation of CMS
- Values still subject to change, but the plots give useful information:
  - plot scale gives an indication of the overall rate
  - relative values give rate reductions
  - shapes give indications of which regions of the detector are subject to higher rates
  - understood rate differences between the CMS full and fast simulations

Average hit rates per z module

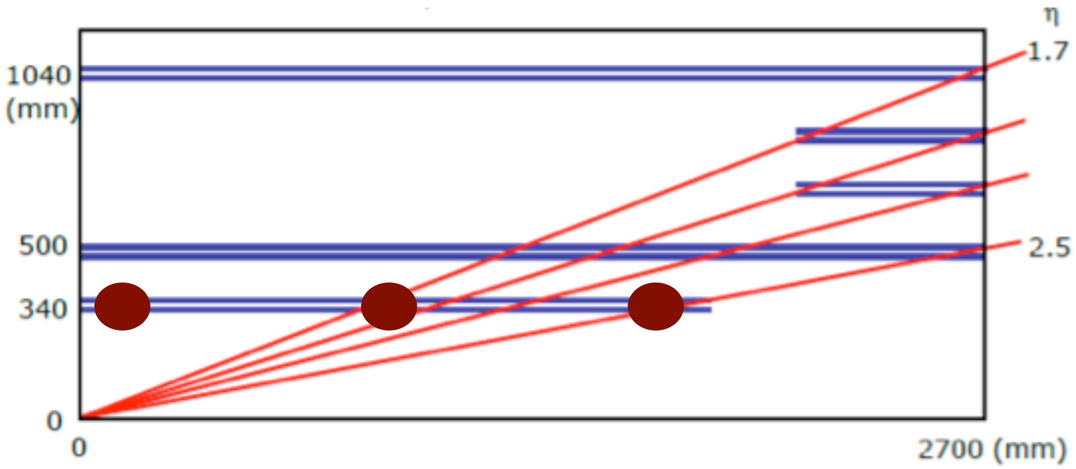


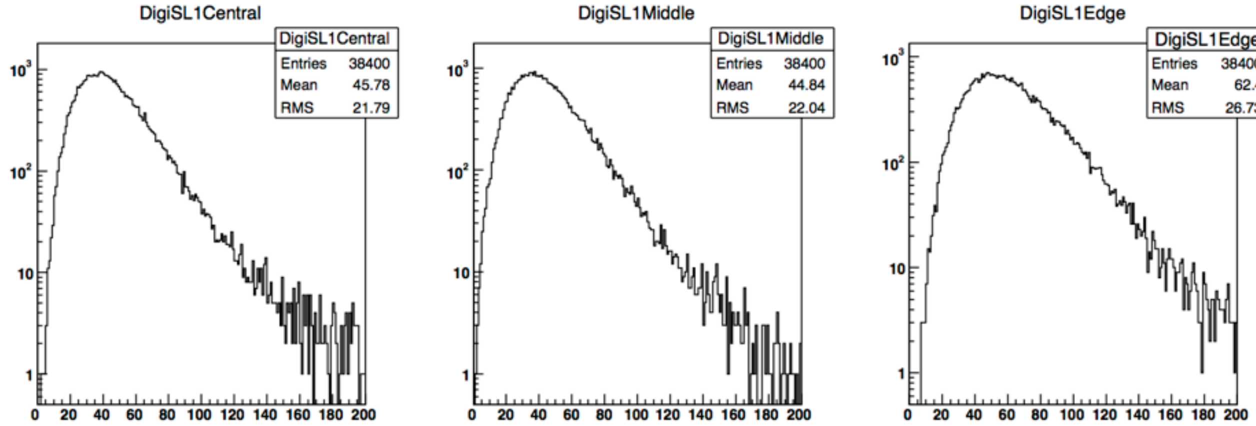


- 16 MHz/cm^2 corresponds to 0.8 hits per cm^2 for 20 MHz bunch crossing rate
- Scale difference between full and fast simulations due mostly to delta rays not included in the fast simulation
- Shape difference at high η values are due to limitations of the fast simulation
- Differences between full and fast simulations significantly reduced with clusters
- Stub rate significantly reduced in the high eta region
  - interaction region requirement to form a stub

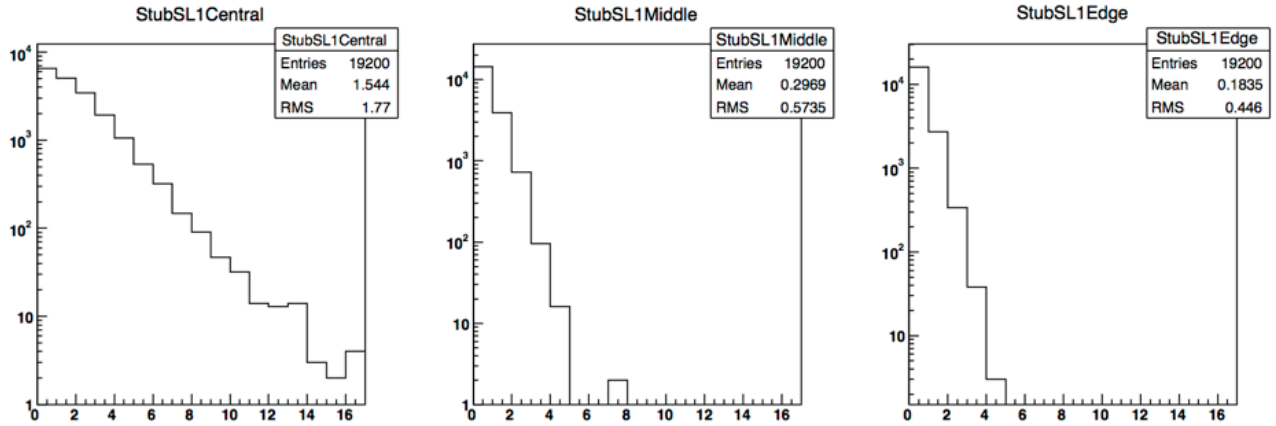
Local hit fluctuations

- Not only average rates, but also hit fluctuations - and tails - are important to design read-out chips
- We study fluctuations in different areas and regions of the detector
- Example: three 100cm^2 modules in layer 1





Digi hits in a module



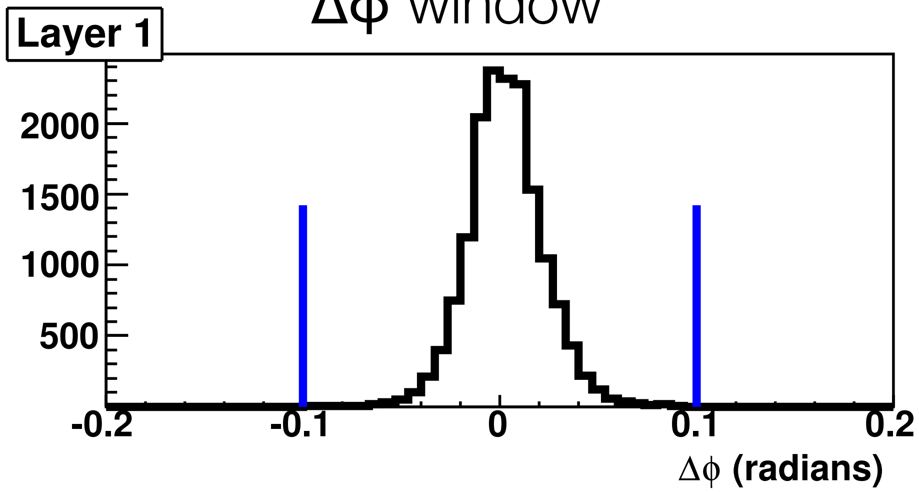
Stubs in a module

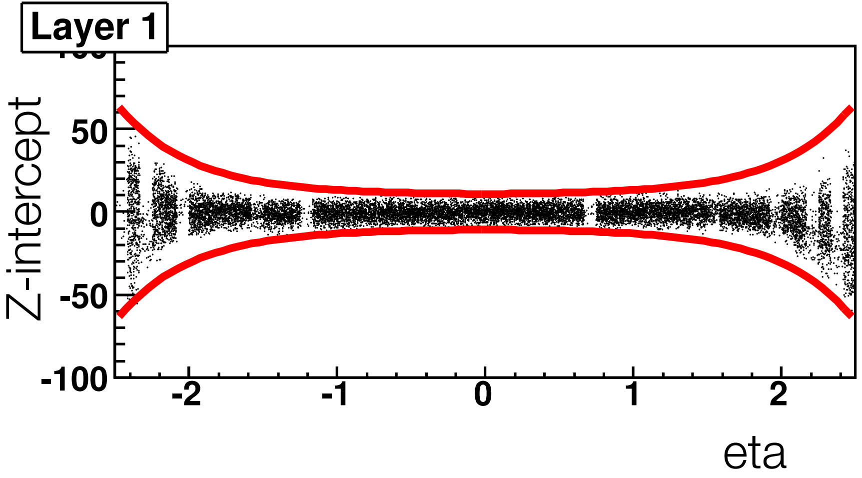
Tracker + ECAL matching: electron example

(study conducted by Laura Fields, former Cornell University post-doc)

- First attempt to match ECAL Level 1 electrons with tracker stubs
- Level 1 electrons are matched to two or more stubs in different tracker layers
- Algorithm to match an electron with a stub:
  - $\Delta\phi$ between a stub and the projected electron trajectory
  - Z-intercept of line between electron and stub on r-z plane
  - similar algorithm to match stubs on different tracker layers

- Study performed on 200PU with the fast simulation only - so far

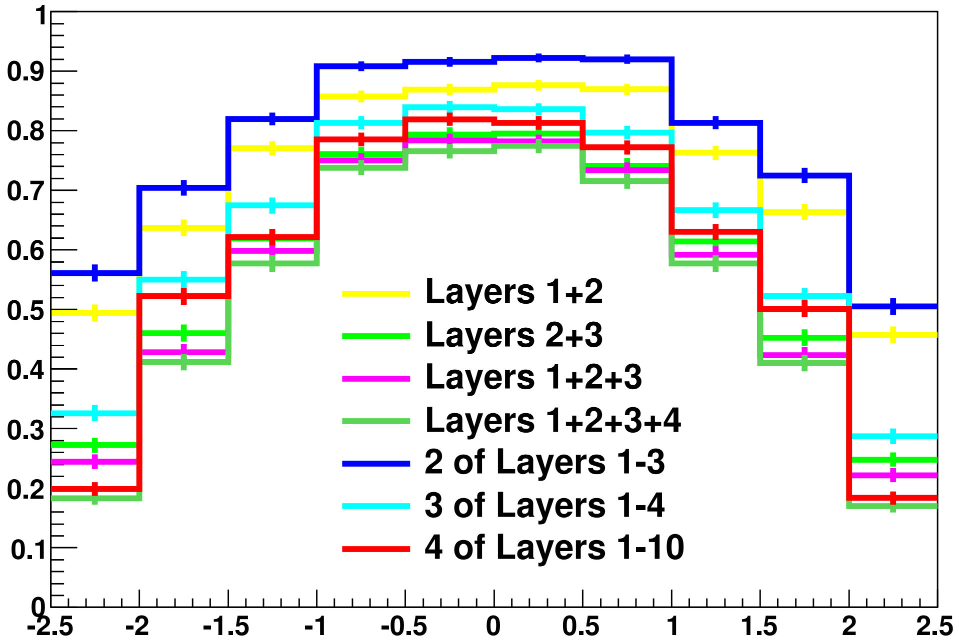

$\Delta\phi$ window


Z window

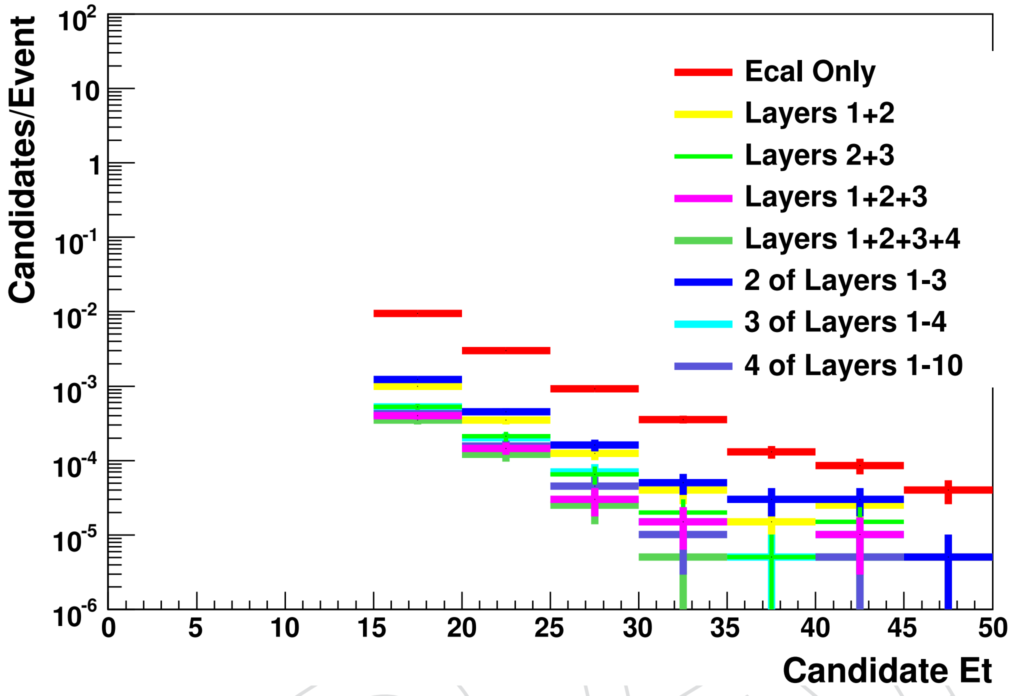
Results from electron study

electrons
sample
Fast sim

Efficiency vs eta



Rates vs E_T

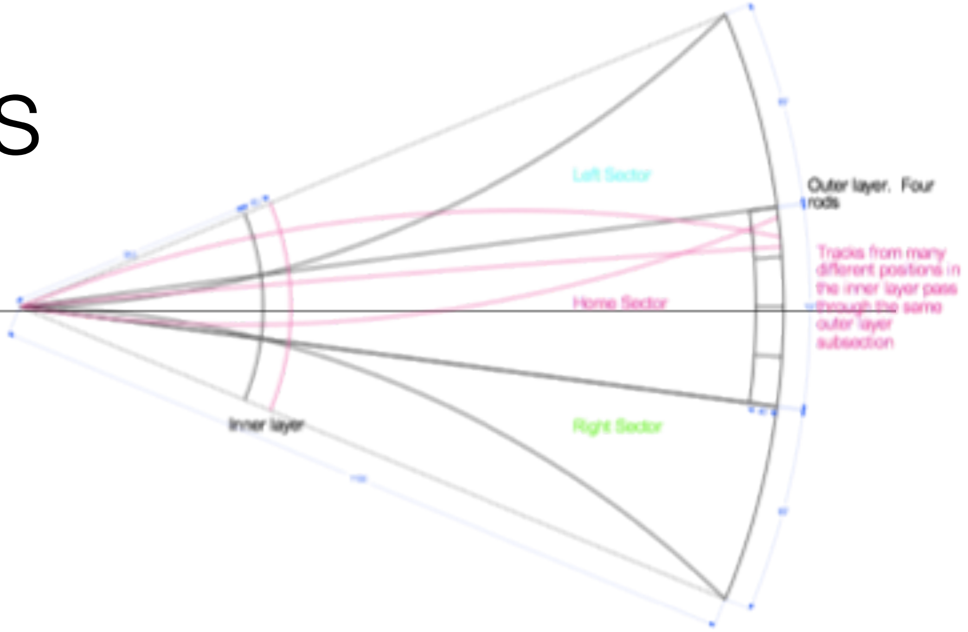


200 PU
Fast sim

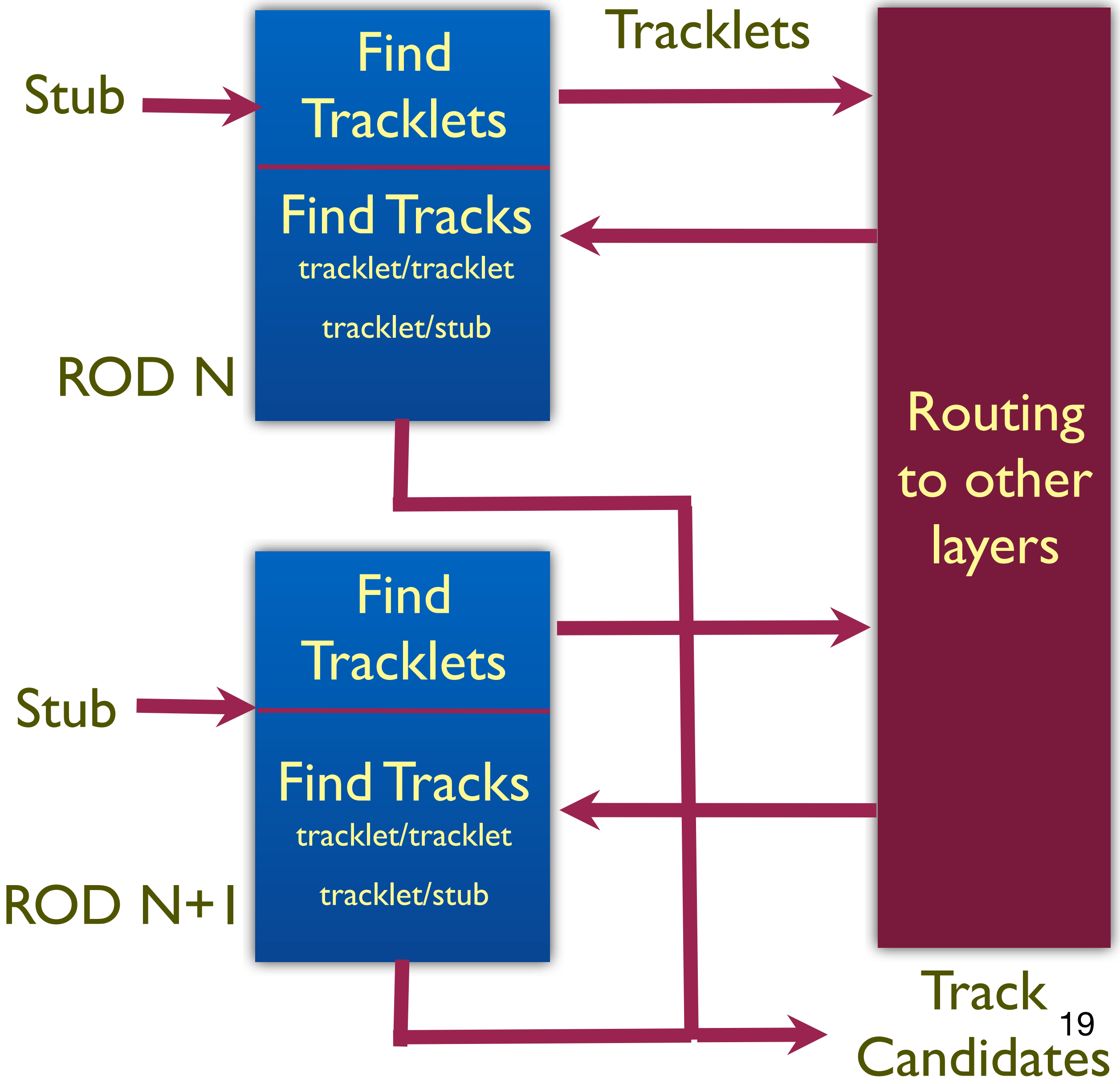
- Significant rate reduction by adding stub information
- Highest efficiency configuration: electron + stubs in two of the first three layers
- Efficiency reduction at higher eta
  - due mostly to material effects
  - usage of fast sim - need to update by using the full CMS simulation
- This study is still preliminary for a quantitative interpretation

Off-detector processing: tracklets

(Ron Lipton, Marvin Johnson)



- The local design minimizes data transfer and interconnection complexity
- Input FPGA finds tracklets from stubs within a rod and finds destination rods in other layers
- Stubs are retained in “home” rod for matching with incoming tracklets
- Tracklets are routed to destination FPGAs where they are combined with other tracklets and stubs to form track candidates
- Resulting track candidates are sent out and possibly redundancy removed
- Only tracklets and tracks are formed across rods


Stub

Find Tracklets

Tracklets

Find Tracks

tracklet/tracklet

tracklet/stub

ROD N

Routing to other layers

Stub

Find Tracklets

Find Tracks

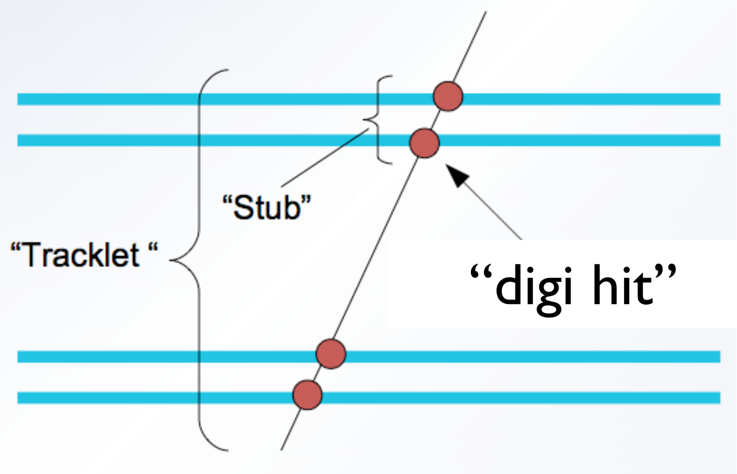
tracklet/tracklet

tracklet/stub

ROD N+1

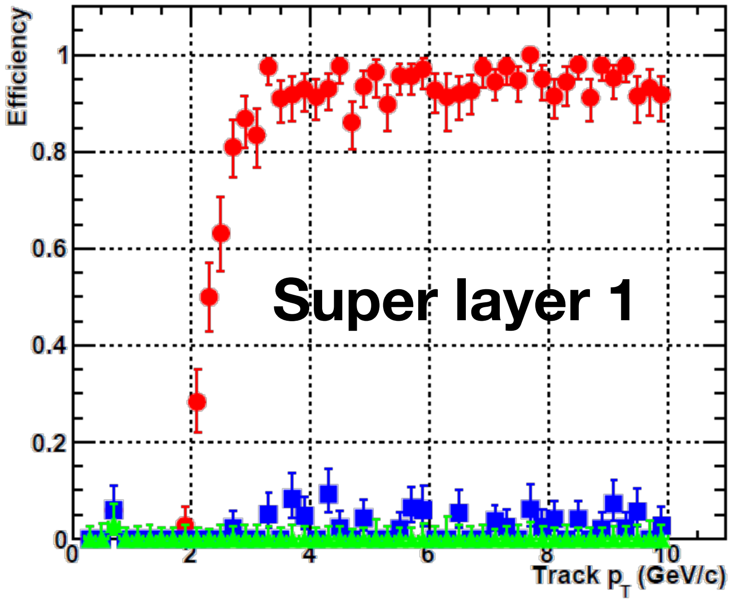
Track Candidates

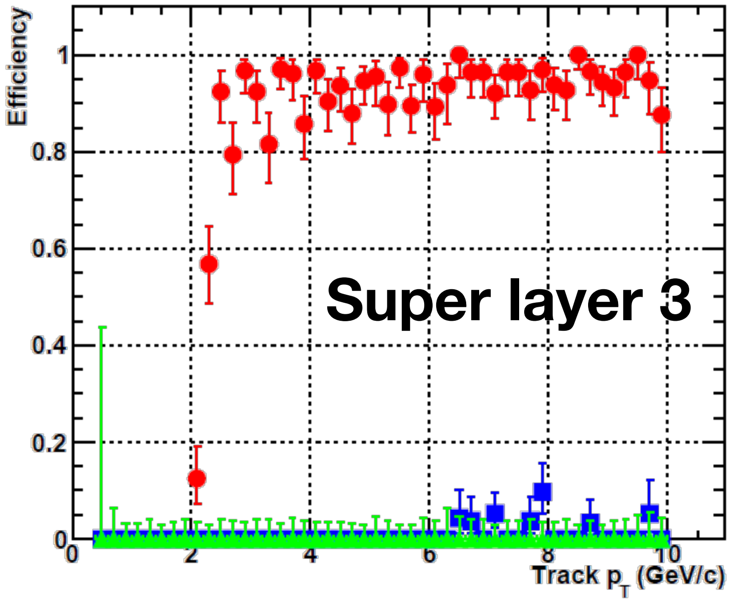
Tracklet formation



- Our tracklet formation algorithm is similar to the algorithm for stub
- Tracklet direction fully constrained if using vertex information
- Tracklet p_T obtained by a fit (two stubs + beamspot)
- Tracklet production efficiency in the first and last Super Layers
  - single muon events
- Reminder: stub production threshold set to 2 GeV/c
  - sharp tracklet production efficiency

At least 1 tracklet
At least 2 tracklets
At least 3 tracklets

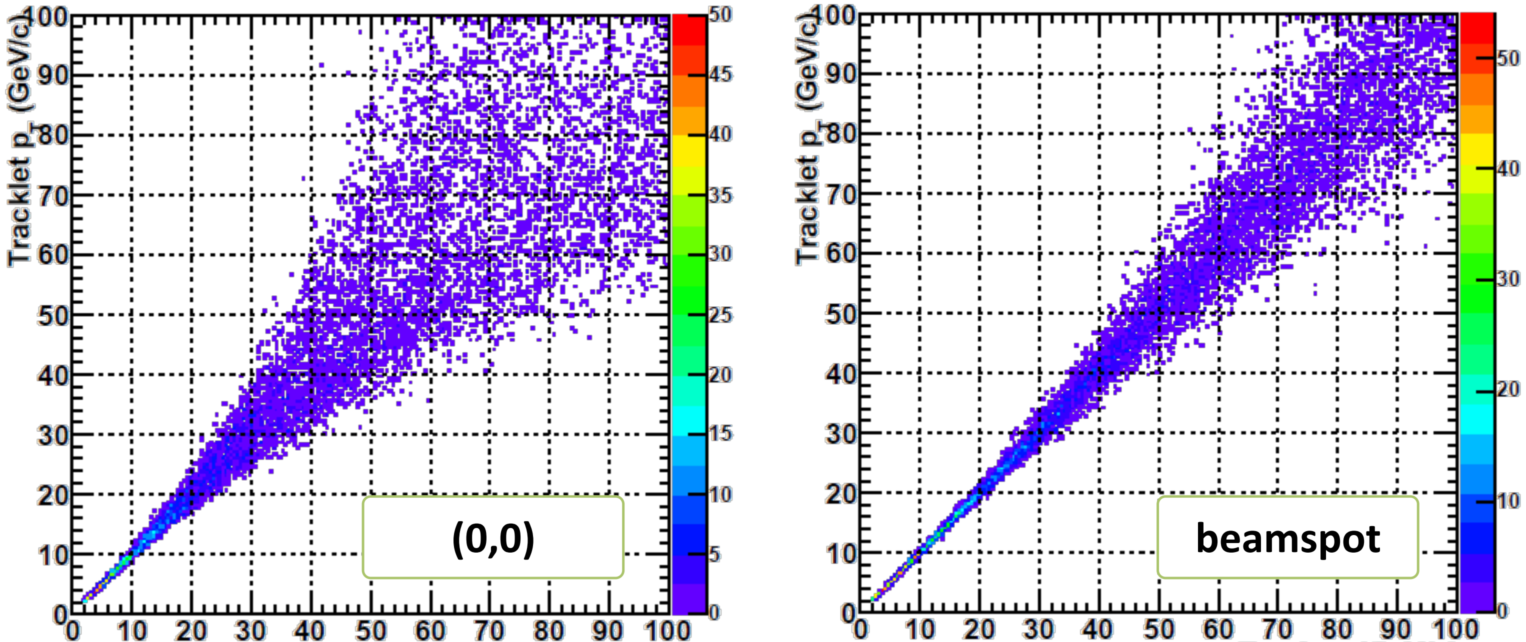


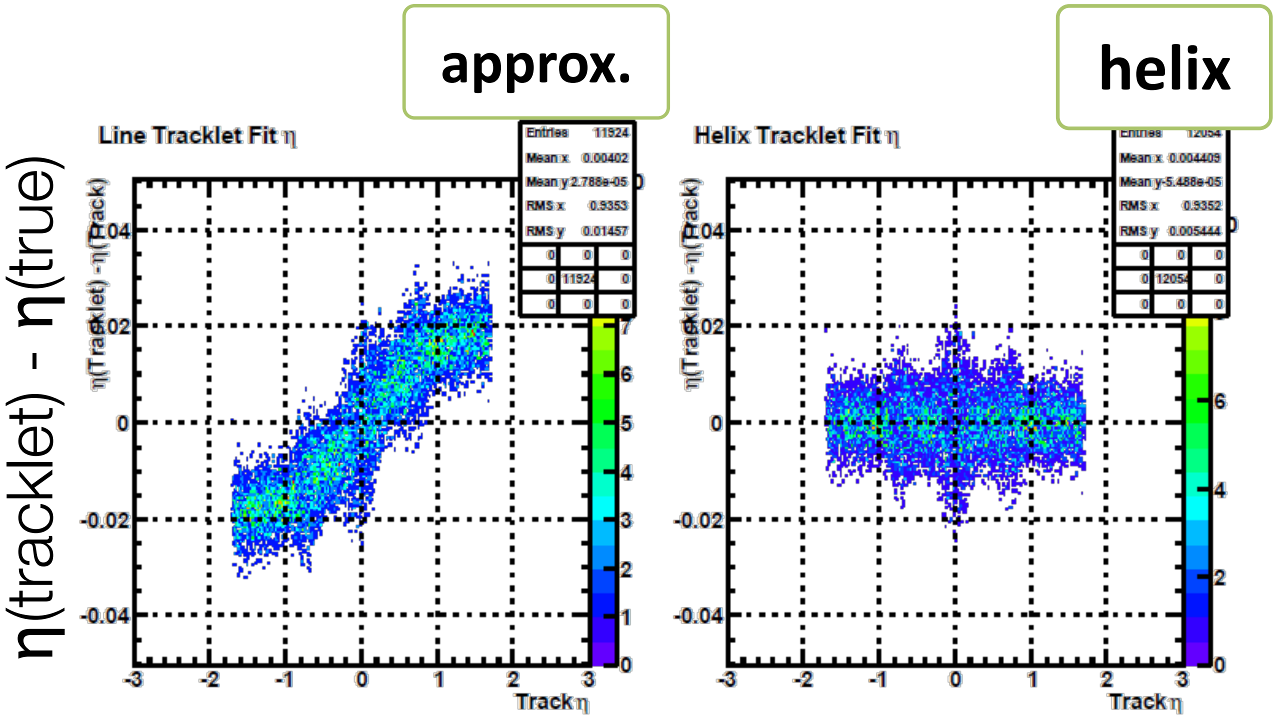


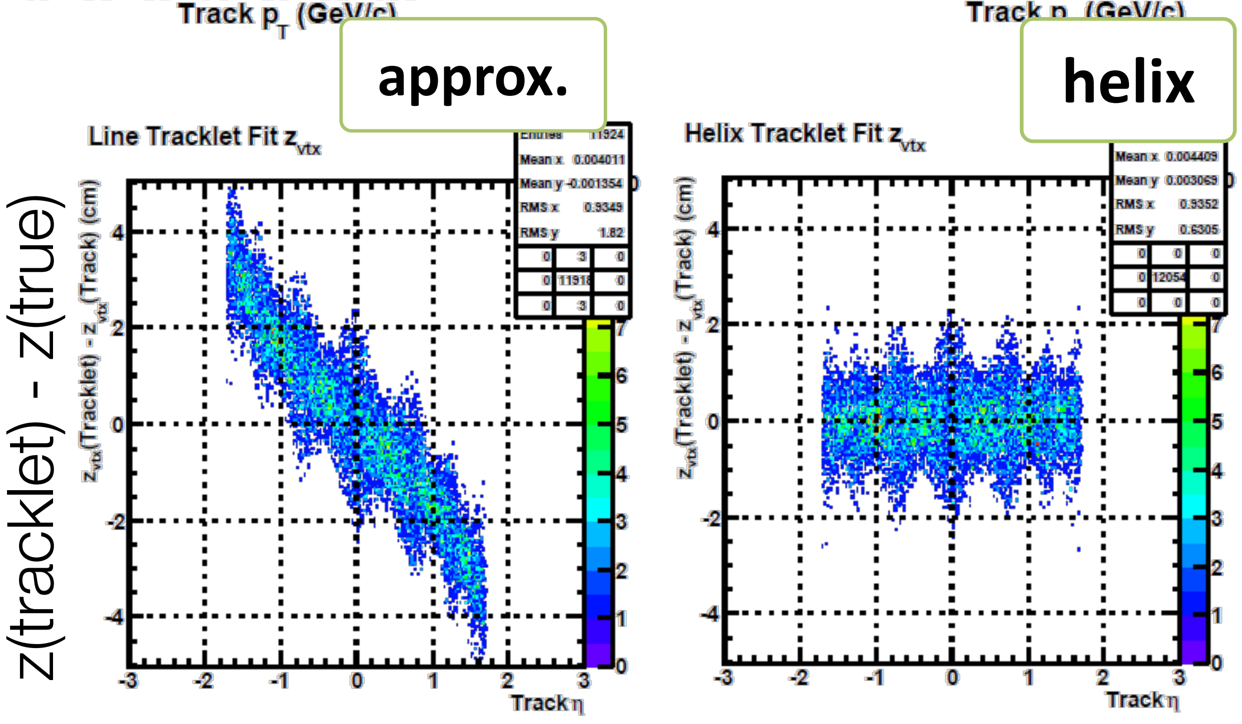
Corrections to tracklet algorithm

- Initial attempt:
  - vertex in (0,0)
  - back-propagation to vertex with a straight-line
- Straight-line fit significantly biassed in the first layers
  - replaced by full helix fit

Beamspot correction

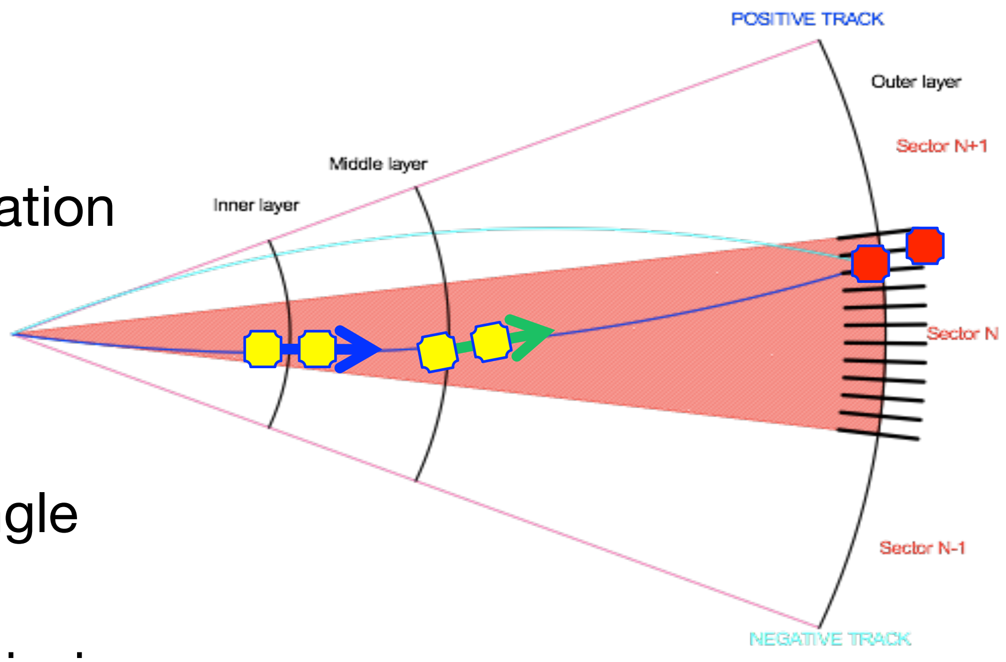






Corrections not really feasible in hardware

Build L1 tracks from tracklets

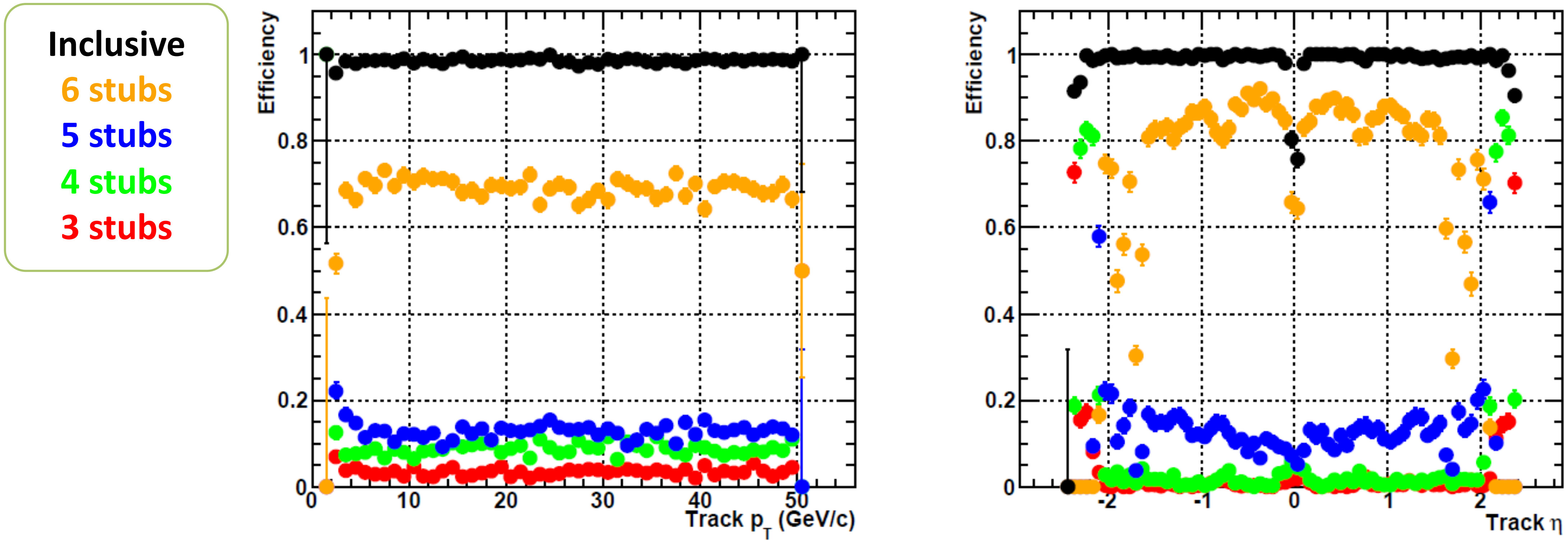


- Goal to minimize the track search region
  - to reduce number of equations for hardware implementation
- L1 tracks built by propagating tracklets to other two Super Layers
  - Reduce inefficiencies by matching a tracklet even to single stubs instead of tracklets on the other Super Layers
    - find best possible track in case some sensors are missing
  - drawback: could find the same track three times
- define Δφ and Δz tables for tracklet-stubs matching for different p_T
- beamspot correction is also needed

- Apply L1 track fit to obtain track p_T (still preliminary)

L1 tracks

- L1 track production efficiency from the first Super Layer (R=32-36 cm)
- We notice some effects due to pixel granularity and trajectory approximation
  - requires further development
- L1 track algorithm is still preliminary and not very realistic
  - **But L1 track objects are already available for studies to match tracking trigger object to the muon and calorimeter triggers!!**



Conclusions

- Adding tracker information at Level 1 triggering is very important for rate reductions
    - A new tracker is required
    - We are studying possible new tracker geometries
- Add momentum information to Level 1 by using pT modules
    - Studying different topologies, sensors, chips...
- We are testing and comparing several ideas for building Level 1 tracks
    - only sequential scheme presented here (several algorithms available)
    - all of them show promising significant rate reductions
- Algorithms for basic primitives quite well tested and established
- Currently work in progress
    - add flexibility to the simulation code (e.g. allow different geometries, topologies)
    - studying how to improve algorithms for design of Level 1 tracks