Status of SiStripHitResolution studies

D. Gelé, (Strasbourg)

(July 18, 2018)

Outline

- Results for 2018 data with Nico's code
- Status of HitRes code transfer into CalibTree framework

Results for 2018 data(1)

- Use of HitRes code provided by Nicola de Filippis (thanks!)

(pair method, some minor changes in the code due to depreciated macros)

Configuration:

CMSSW_10_1_7 + GlobalTag = 101X_dataRun2_Prompt_v11

2 StreamExpress data samples:

/StreamExpress/Run2018A-SiStripCalMinBias-Express-v1/ALCARECO

/StreamExpress/Run2018B-SiStripCalMinBias-Express-v1/ALCARECO

corresponding to 14.7 fb-1 and 7.1 fb-1 (from brilcalc.py with json file, no normtag file)

Processed by means of CRAB

Results for 2018 data(2)

- Selection cuts:

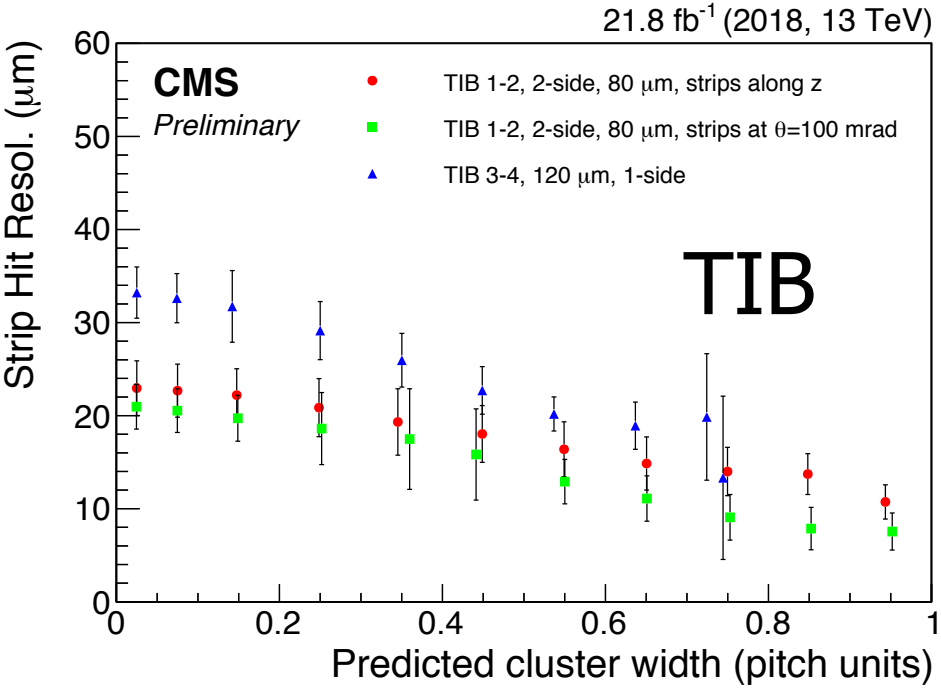
Trajectory cuts: p_T > 3 GeV
$\chi 2$ Prob >= 0.001
Found n. recHits>=4 (including pixels hits-> t.b.c.)

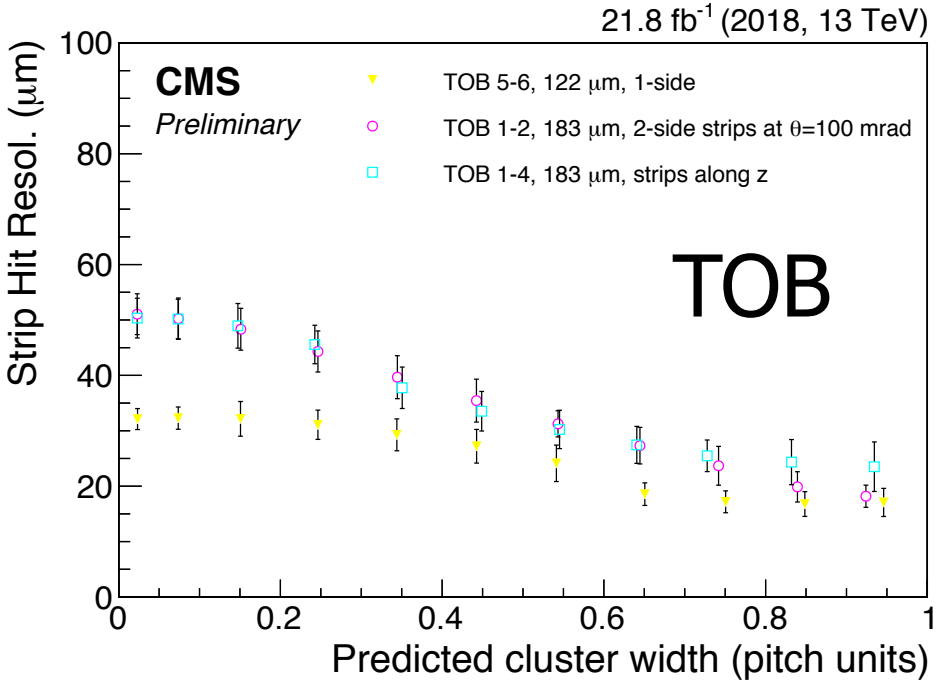
then selection of hits in overlapping module of the same layer with

Pair requirements: At most 4 strips large cluster width
Clusters are of the same width in both modules
Clusters are not at the edge of the modules
Predicted path < 7cm (distance of propagation from one surface to the next)
Error on predicted Δx <=0.0025 cm(Δx: difference between predicted backward and forward traj. states from 2 overlapping sensors)

Results for 2018 data(3)

- Presentation similar to 2017 results for comparison:



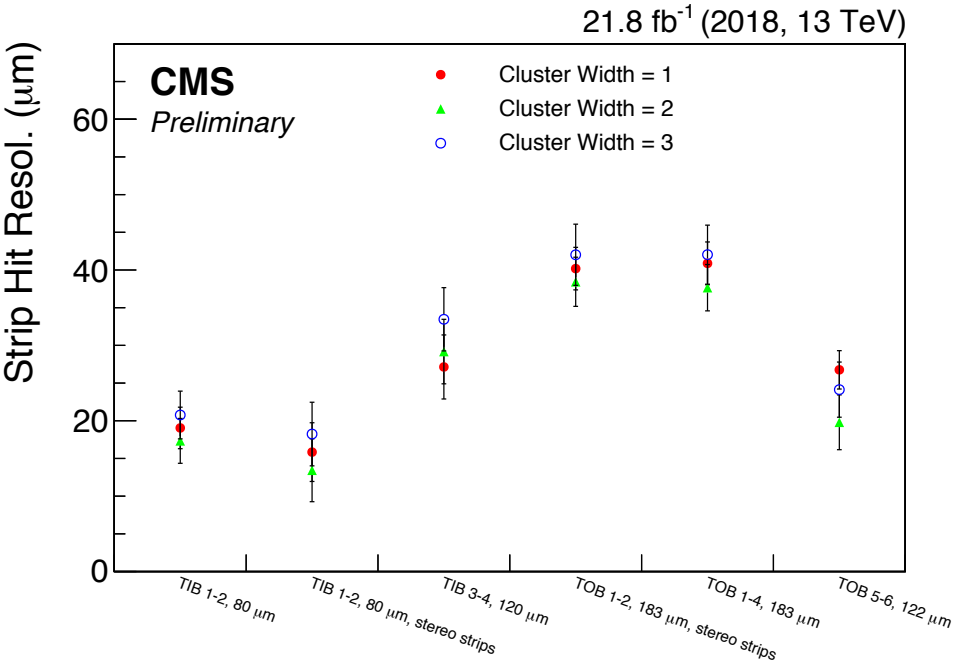


SiStrip resolution by selection of hit pairs in overlapping 2-sided or 1-sided TIB or TOB modules of the same layer as a function of the expected cluster width (in pitch sensor units)

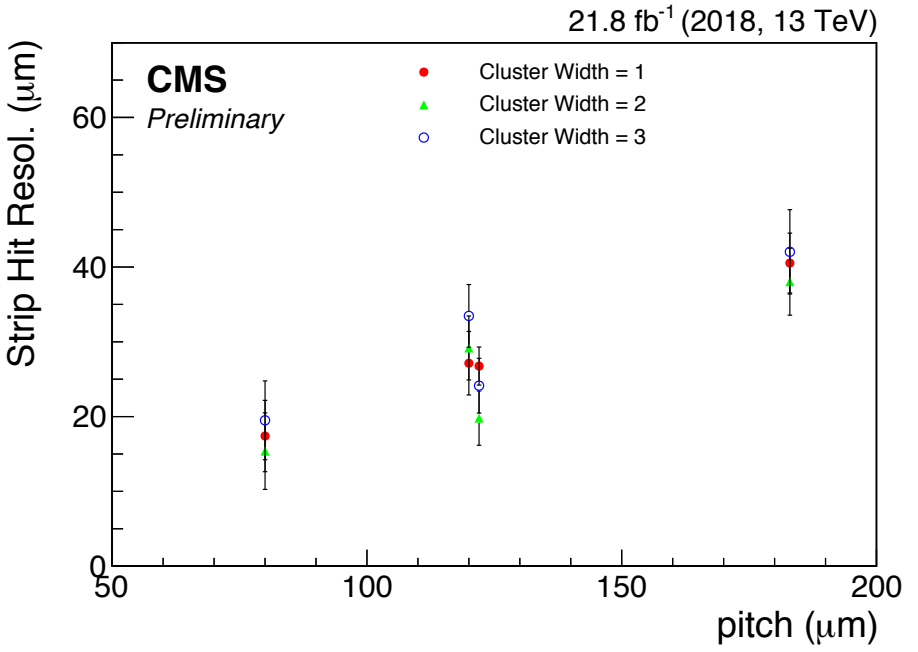
=> Very similar to 2017 results

Results for 2018 data(4)

- Presentation similar to 2017 results for comparison:



SiStrip resolution with pair method for different types of overlapping sensors and cluster width (in number of strips)



SiStrip resolution with pair method in overlapping sensors (with different pitch values) and for different cluster widths (in number of strips)

=> Very similar to 2017 results

Code for SiStripHit Resolution in CalibTree(1)

- Goal: implement Nico's code into CalibTree framework:

This is done in the HitEff.C code of SiStripHitEfficiency branch (in a totally independent way of the hit efficiency code itself)

Compared at the level of a few events with Nico's code:

Same features for refitted trajectories (and identical values for ntuple hitres content) BUT more trajectories in Nico's code wrt the new one (as if there was an additionnal cut/prescale ? -> not seem to be in refit config file) => not found the origin of this discrepancy

Code for SiStripHit Resolution in CalibTree(2)

```
Starting analysis, nrun nevent: 315800 752591021
Pt traj = 0.873868
Pt traj = 0.878073
Pt traj = 0.881786
Pt traj = 0.884597
Pt traj = 0.884975
Pt traj = 0.882551
Pt traj = 0.884227
Pt traj = 0.888749
Pt traj = 0.889073
....
Pt traj = 0.913083
Pt traj = 0.917426
Pt traj = 0.915366
Pt traj = 3.57021
Pt traj = 3.57207
Pt traj = 3.57362
....
momentum 0.991502
track_trackChi2      0.0866089
track_trackChi2_2    0.0866088
track_eta           -1.72062
momentum             7.57884
numHits              20
trackChi2            0.00989575
detID1               436266440
ptich1               0.0183
clusterW1            6
expectedW1           0.509913
atEdge1              0
simpleRes            0.0135903
detID2               436266472
clusterW2            2
expectedW2           0.22607
atEdge2              0
pairPath            -3.28111
hitDX                8.86621
trackDX              8.85336
trackDXE             0.0007675
trackParamX          4.32746
trackParamY          2.94283
trackParamDXDZ      -0.00428619
trackParamDYDZ      -0.262439
trackParamXE         0.00242244
trackParamYE         0.0276596
trackParamDXDZE      0.000223308
trackParamDYDZE      0.000886589
....
Pt traj = 0.988768
Pt traj = 0.989966
Pt traj = 0.995559
Pt traj = 0.991502
Starting analysis, nrun nevent: 315800 751544428
```

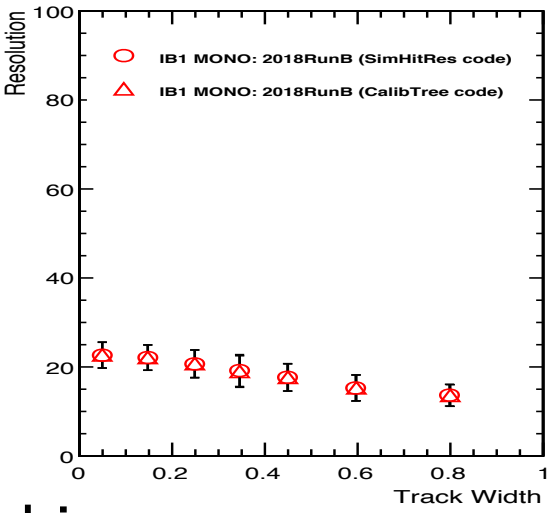
Nico's code

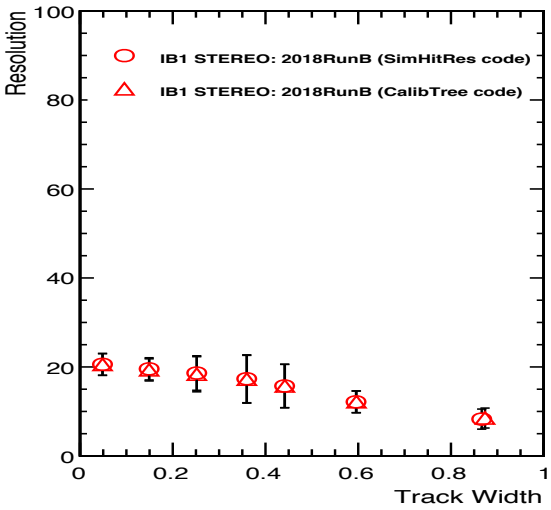
```
Starting analysis, nrun nevent: 315800 752591021
Pt traj = 0.873868
Pt traj = 0.878073
Pt traj = 0.881786
Pt traj = 0.884597
Pt traj = 0.884975
Pt traj = 0.882551
Pt traj = 0.884227
Pt traj = 0.888749
Pt traj = 0.889073
....
Pt traj = 0.913083
Pt traj = 0.917426
Pt traj = 0.915366
Pt traj = 3.57021
Pt traj = 3.57207
Pt traj = 3.57362
Starting analysis, nrun nevent: 315800 751544428
```

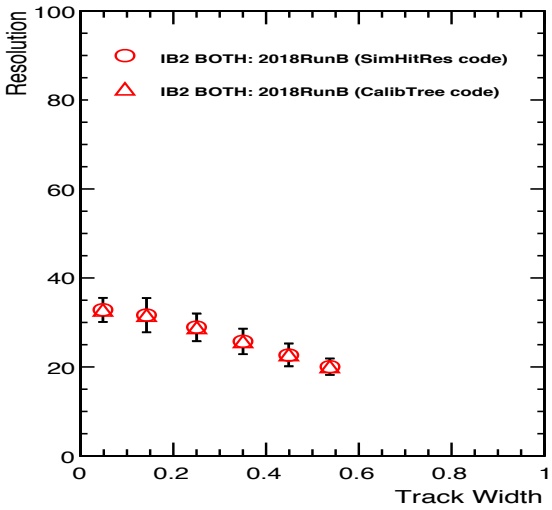
CalibTree code

Example: Scan of one event

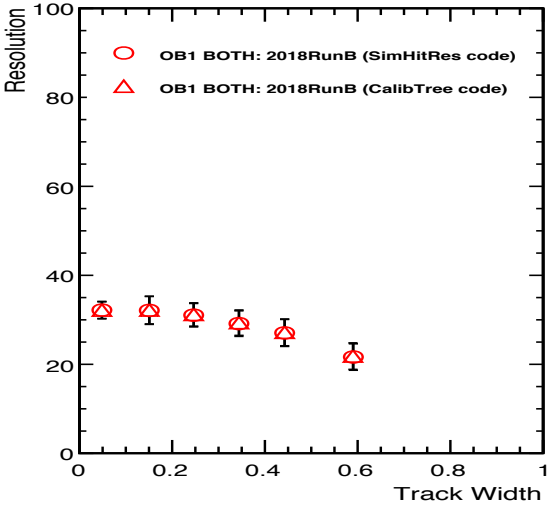
Code for SiStripHit Resolution in CalibTree(3)

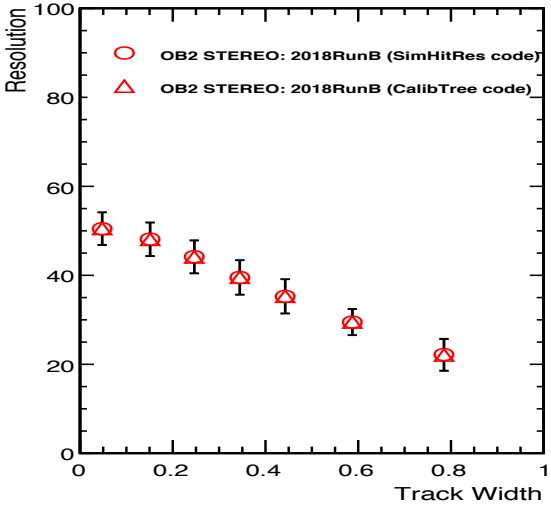


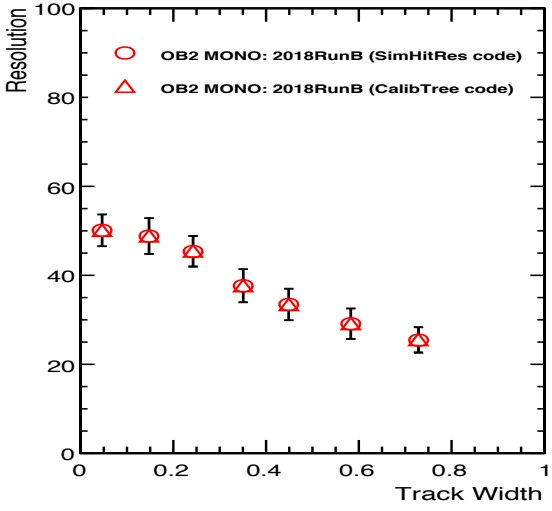




No bias







Code for SiStripHit Resolution in CalibTree(2)

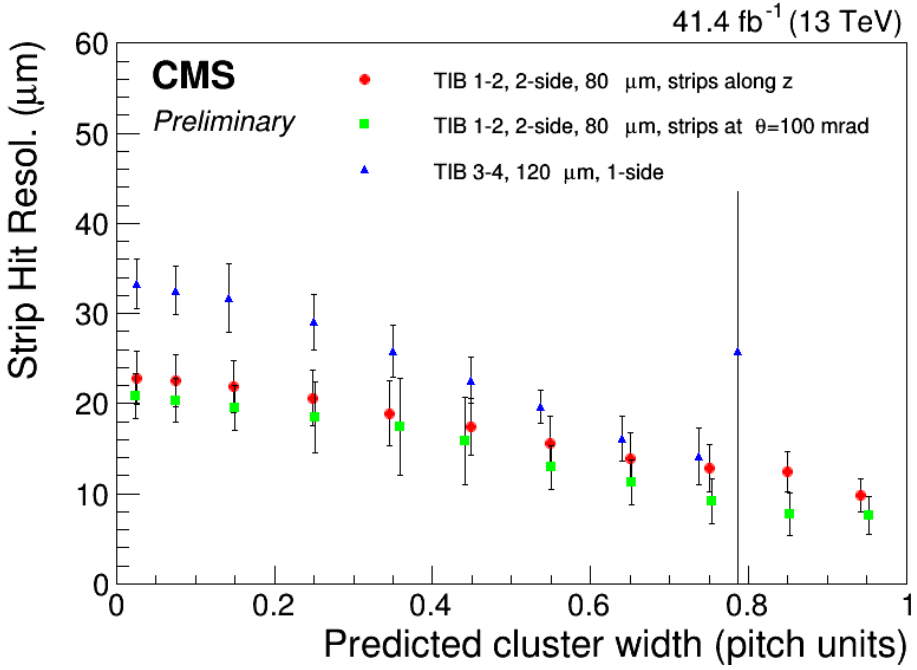
- Status:

When keeping only the hit pair candidates (with a Pt cut for traj>3GeV): increase of <2.5% (size of ntuple) and about 10% increase for time computing.

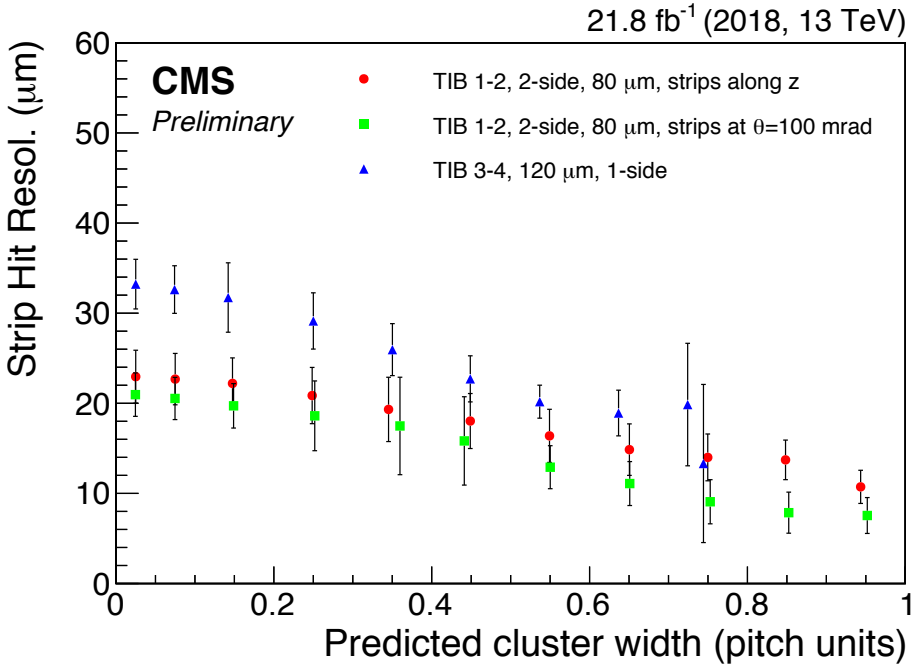
Still some cleaning/commenting to be done and cuts to be given as input in .py file

BACK-UP

2017 vs 2018 comparison

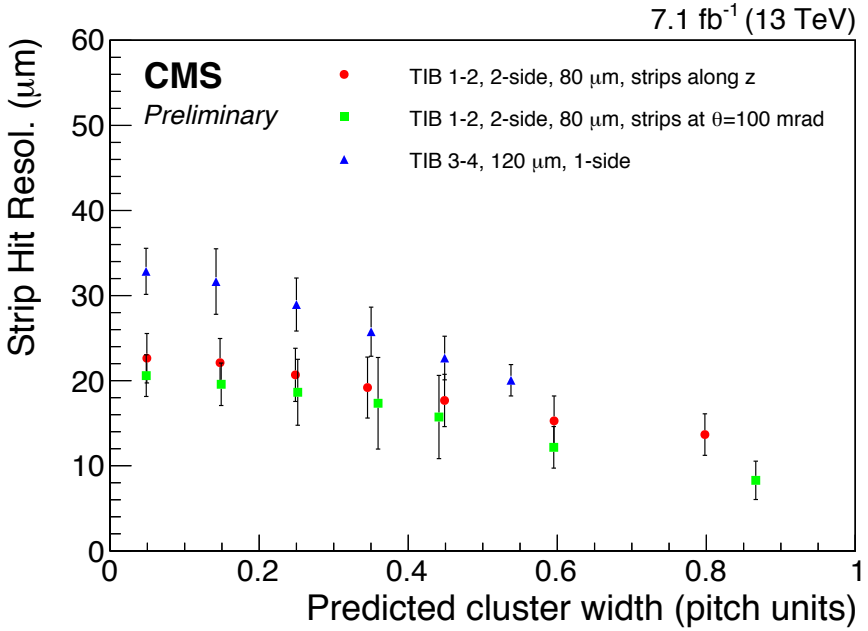


2017

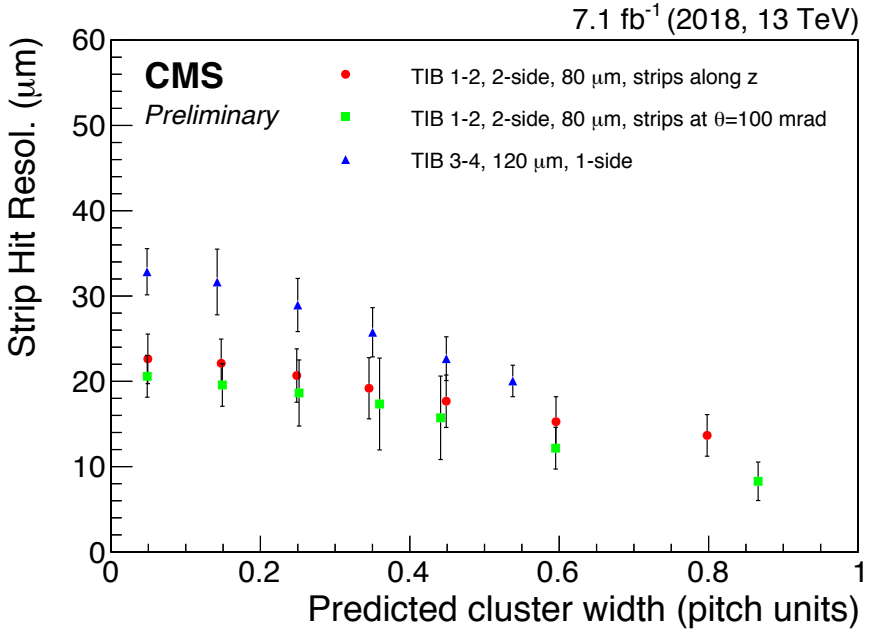


2018

2018 Nico's vs CalibTree codes comparison



Nico's code



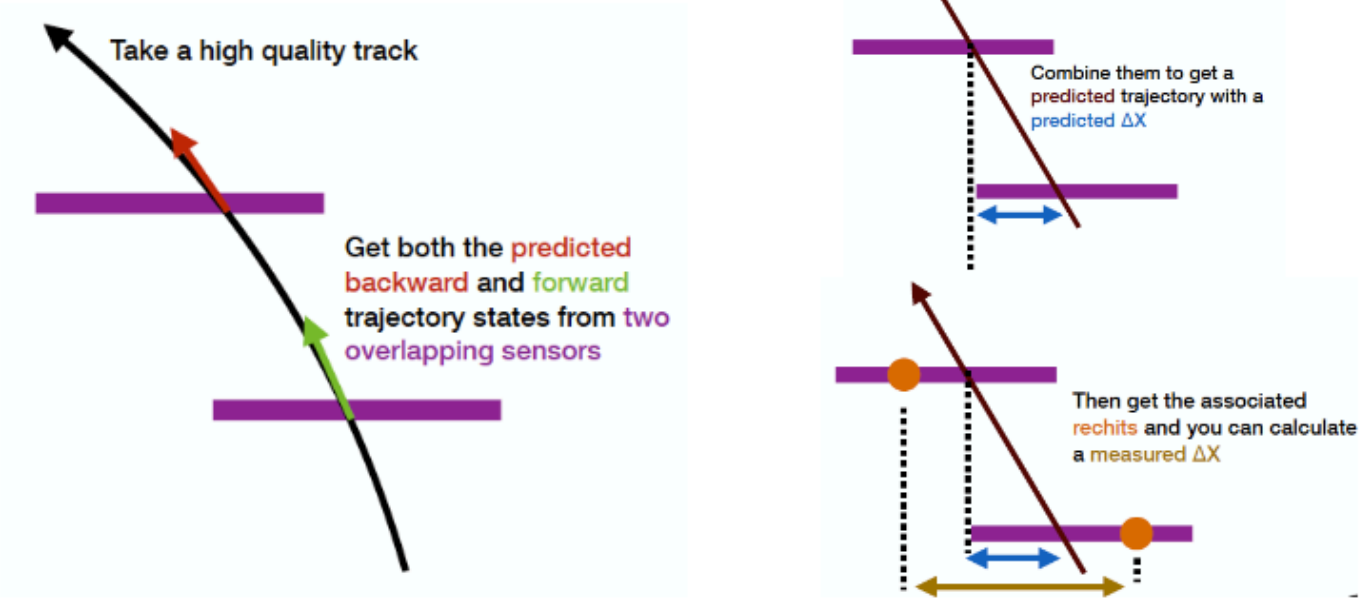
Calib Tree code

From Nico's presentation:

Strip Hit Resolution: "pair method"

Tracks are selected with high quality criteria.

Along those tracks two hits on overlapping modules of the same layer



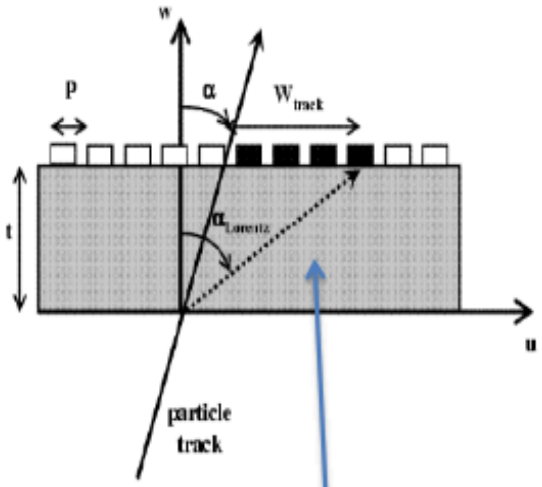
measured ΔX - predicted ΔX ≃ the sum of two residuals

From Nico's presentation:

Hit position: is the cluster baricenter corrected with the shift due to the Lorentz drift.

Hit position resolution:

- in the measurement frame is set to be **1/√12** in both 2D coordinates for each hit (uniform distribution)
- in the local frame the error is **pitch/√12** for the precise coordinate and **(strip length)/√12** for the coarse coordinate.



- It has been measured using MC truth and is parameterized as a function of the reconstructed cluster width (actual size of the cluster, in number of strips) and the expected cluster width.

- The expected cluster width / track width is the track projection on the precise coordinate in the measurement frame corrected with the Lorentz shift, in units of the strips.

From Nico's presentation:

Resolution:

Simple:

- Take re-fit of track trajectory w/o RecHit
- RMS(Track X - RecHit X)

Pair method:

- Use two hits on overlapping modules