CA On GPU: Strip Extension

Bi-weekly WP2 meeting

22 October 2024

Brunella D'Anzi on behalf of the CA extension group

TRacKing CA+Strips: Motivation

- The Cellular Automaton (CA) is an algorithm designed for parallel architectures, can be used either as the **main track finder algorithm** (eg. at **HLT**) or as a **seeding step** to another algorithm (eg. the Kalman filter method used in **offline reconstruction**).
- Cellular Automaton algorithm is applied only to **CMS Pixel detector** currently.
- Goal: measure **CA performance** including the Strip information (in a smart way)

Motivation of CA extension to use pixel+strips:

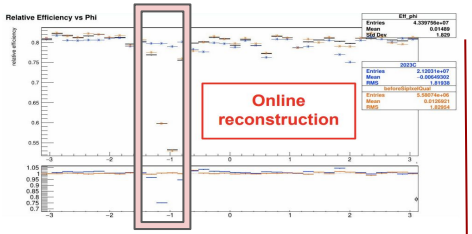
- Momentum resolution improvement
- Fake tracks reduction
- Recover "pixel off" failures

Example: Layers 3 and 4 of BPix-Sector 7 have lost QPLL lock

- recovery attempts have been unsuccessful this config stays in 2024 as well
- 27 associated modules have been masked, corresponding to a region spanning ~24° in ϕ (around ϕ=-1)

Observed drop in tracking efficiency

- offline tracking almost unaffected [ratio of D/C : ~20%]: iterative procedure relying on different sets of seeding naturally recover the inefficiency
- online tracking [ratio of D/C : ~50%]: relies only on pixel triplets and quadruplets
  - needed a dedicated extra pixel pair step for recovering the localized inefficiency



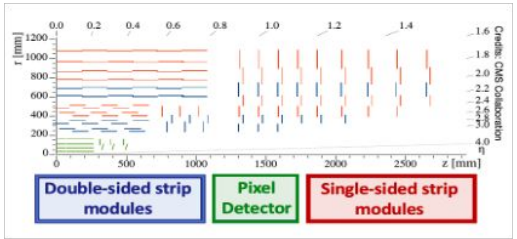
Momentum scale resolution

- Relative pT resolution is worst for low pT tracks because of multiple scattering
- Resolution degrades at high pT because tracks are less bend in the magnetic field
- Resolution is best for central tracks and for tracks of O(10) GeV

Impact parameter

- It degrades for forward tracks

  The addition of TIBs + TIDs should help on this!



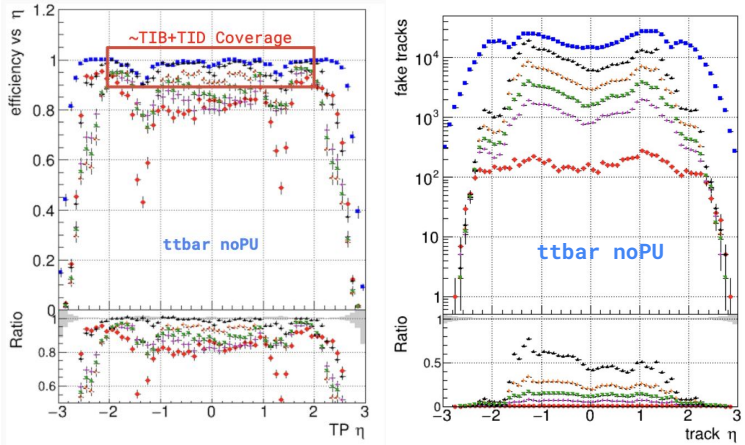
TRacKing CA+Strips: Progresses

We were starting from a Alpaka working version with degraded performance for **pixelStripQuadruplets** w.r.t. the **basic pixelTriplets tracks** (default now in CMS @HLT) in **high eta region**.

Tasks

1. Include a rebase to 14_0_15 ✅
2. Usage of the CA autotuning package to work with strips. ✅
3. Test different failure scenarios. ✅
4. Open to pixelTriplets to have a compatible pixelTriplets vs (pixelTriplets+pixelStripQuadruplets) comparisons. ✅
5. Decreasing of fake rates ✅
6. Ready HLT menu including Strips and on-going timing measurements (for now global strip unpacking)
7. Started moving geometry structures to be read at runtime for pixel-only, targetting 14_2_X to avoid HIon conflicts (opened PR45421).

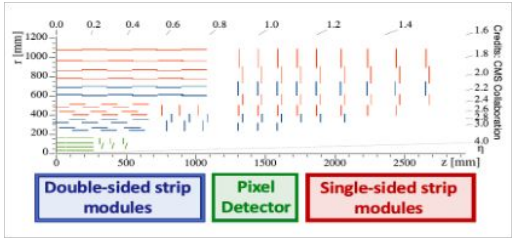
Starting point from the July Patatrack hackathon



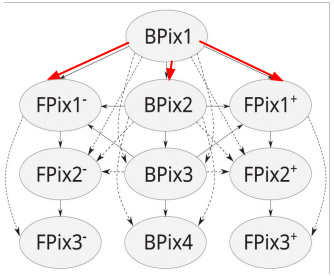
In the following results for pixelTriplets+pixelStripQuadruplets:
wp1: higher efficiency, higher fakes+duplicates (according to optimizer)
wp2: lower efficiency, lower fakes+duplicates
Results on offline reconstruction for the first slides

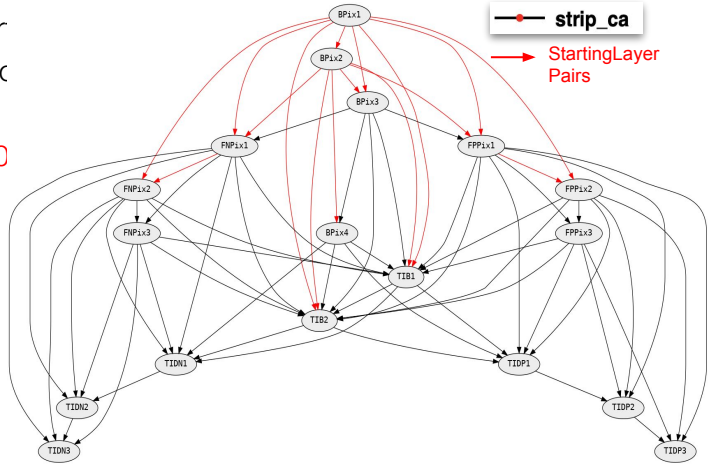
- Pixel Triplets
- Pixel Quadruplets
- Pixel + Strip Quads (4p,3p+1s,2p+2s) (selected wp).
- (Other colors = different strip WPs.)

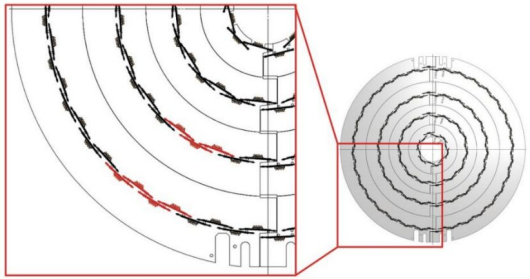
TRacKing CA+Strips: Graphs



- CA Implemented using Alpaka.
- Only **double-sided strip modules** are considered
- It requires a list of layers and their pairings (a graph of all the possible connections between layers)
- At least **pixel triplets** OR **pixel+strip quadruplets to build tracks**
- CA uses a layer adjacency graph:
  - Only pairs of connected layers will be used to build doublets
  - Jump connections allow track reconstruction with missing hits (dashed lines)
  - Definition of **starting layers (solid red lines in the graphs)** to star **ntuple building** requiring a customized Sequential ID to map modules tc layers
  - Added **jumping connections only in pixel holes ranges** (e.g. z in [-28.0 0.])







TRacKing CA+Strips: Scenarios for this study

Presentation from TRK DPG

- 2023D postBPix failure -> Parameter Optimization on tt PU sample (500 evts)

wp1:

```
[0.0006666666698331634, 0.0075165814621699 76, 0.001120536625674629, 0.025168217043074842, 0.05000000198682149, 0.083
33333333333333, 0.4232014167059476]
```

wp2:

```
[0.001030188709485738, 0.007882588743556192, 0.002498967167268065, 0.04850317525835994, 0.05358296209611823, 0.1344
8863622315804, 0.4563450086941161]
```

```
> Parameters to tune:
> > Read in input: ['CAThetaCutBarrel', 'CAThetaCutForward', 'CAThetaCutStrip', 'hardCurvCut', 'dcaCutInnerTriplet', 'dcaCutOuterTriplet', 'dcaCutOuterTripletStrip
'] ( saved in params_to_run.csv)
> Parameter values:
> > Default values:
        -CAThetaCutBarrel: 0.0020000000949949026
        -CAThetaCutForward: 0.003000000026077032
        -CAThetaCutStrip: 0.003000000026077032
        -hardCurvCut: 0.03284072249589491
        -dcaCutInnerTriplet: 0.15000000596046448
        -dcaCutOuterTriplet: 0.25
        -dcaCutOuterTripletStrip: 0.25

> > Low bounds:
> > From input (factor=0.333):
        -CAThetaCutBarrel: 0.0006666666983316342
        -CAThetaCutForward: 0.001000000008692344
        -CAThetaCutStrip: 0.001000000008692344
        -hardCurvCut: 0.010946907498631636
        -dcaCutInnerTriplet: 0.05000000198682149
        -dcaCutOuterTriplet: 0.08333333333333333
        -dcaCutOuterTripletStrip: 0.08333333333333333

> > High bounds:
> > From input (factor=3.000):
        -CAThetaCutBarrel: 0.006000000284984708
        -CAThetaCutForward: 0.009000000078231096
        -CAThetaCutStrip: 0.009000000078231096
        -hardCurvCut: 0.09852216748768473
        -dcaCutInnerTriplet: 0.45000001788139343
        -dcaCutOuterTriplet: 0.75
        -dcaCutOuterTripletStrip: 0.75
```

Params related to triplets building

CA ptmin = 0.55 GeV

- EORun3 worst scenarios

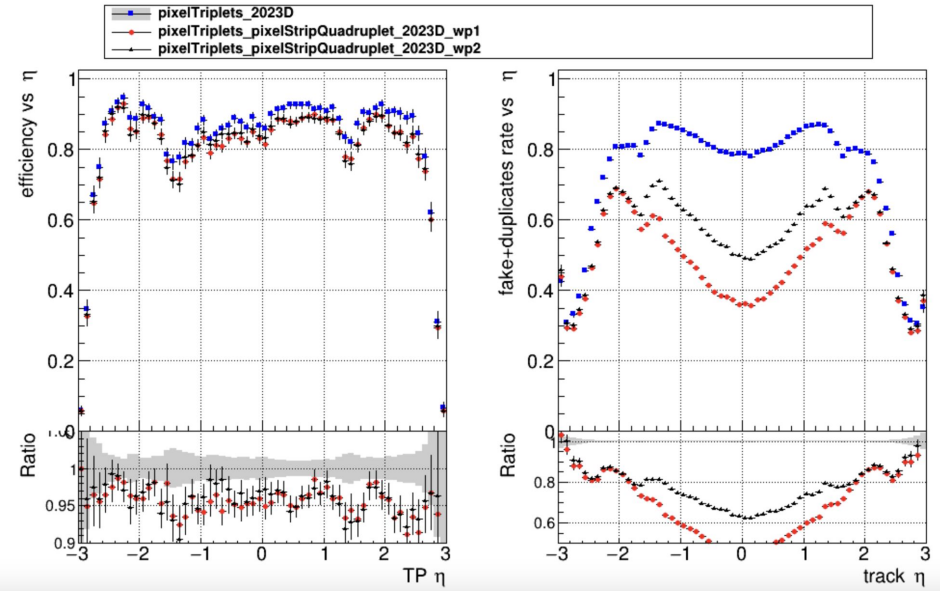
Pixel	Strip	Final
Pix v4	Strip v1	140X_mcRun3_2024_realistic_EOR3_TkDPGv1
	Strip v4	140X_mcRun3_2024_realistic_EOR3_TkDPGv2
	Strip v5	140X_mcRun3_2024_realistic_EOR3_TkDPGv3
	Strip v6	140X_mcRun3_2024_realistic_EOR3_TkDPGv4
Pix v8	Strip v1	140X_mcRun3_2024_realistic_EOR3_TkDPGv5
	Strip v4	140X_mcRun3_2024_realistic_EOR3_TkDPGv6
	Strip v5	140X_mcRun3_2024_realistic_EOR3_TkDPGv7
	Strip v6	140X_mcRun3_2024_realistic_EOR3_TkDPGv8

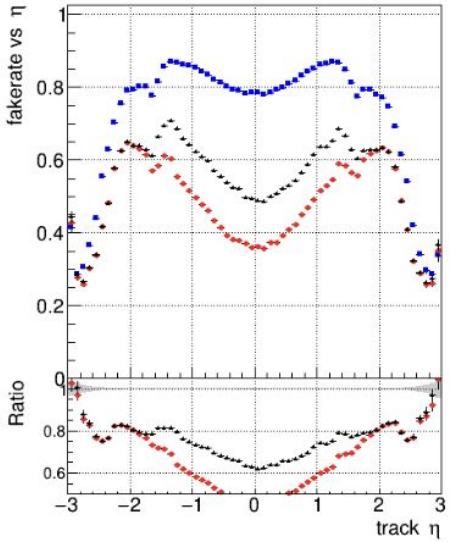
The tags are following

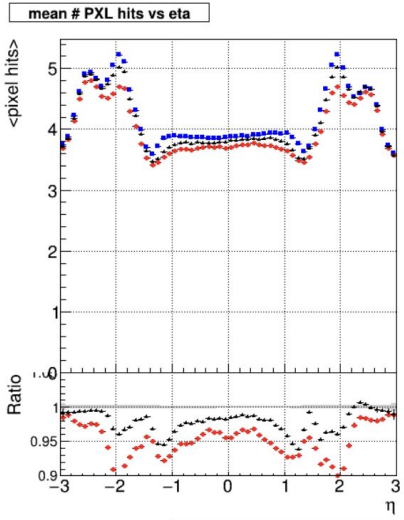
- **Strip v1** - SiStripBadComponents_2024FailureScenario_ThermalRunaway20C
- **Strip v2** - SiStripBadComponents_2024FailureScenario_ThermalRunaway20C_rphiOnly
- **Strip v3** - SiStripBadComponents_2024FailureScenario_ThermalRunaway20C_stereoOnly
- **Strip v4** - SiStripBadComponents_2024FailureScenario_ThermalRunaway25C
- **Strip v5** - SiStripBadComponents_2024FailureScenario_ThermalRunaway25C_rphiOnly
- **Strip v6** - SiStripBadComponents_2024FailureScenario_ThermalRunaway25C_stereoOnly

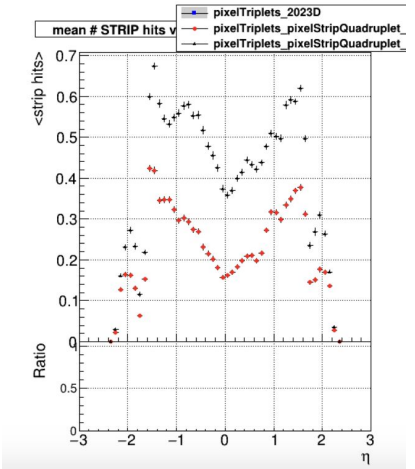
TRacKing CA+Strips: 2023D scenario

- **Efficiency** compatible with pixelTriplets at high eta as well (within relative **5-10%**)
- pixelTriplets+StripQuadruplets have a **lower fake (relative 20-40%)** than pixelTriplets





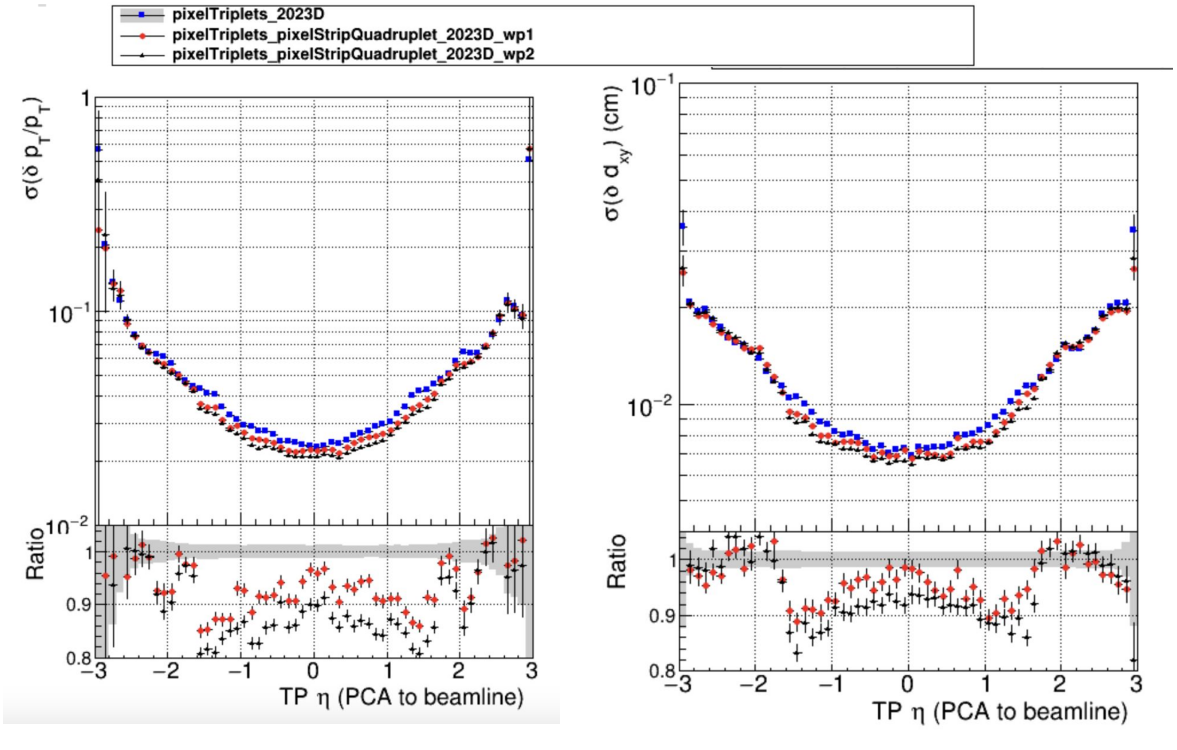




Additional plots here , MTV TrackingParticleSelection cuts

TRacKing CA+Strips: 2023D scenario

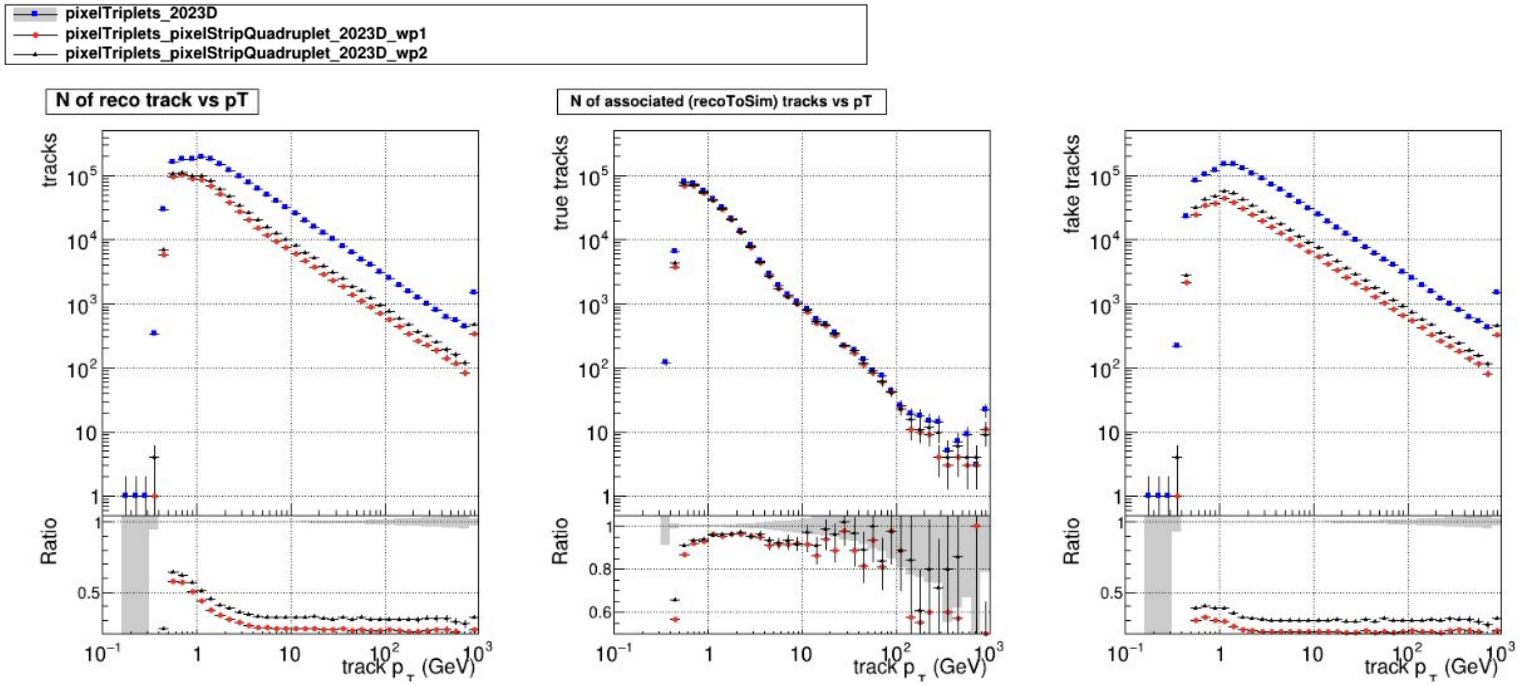
- **Efficiency** compatible with pixelTriplets at high eta as well (within relative **5-10%**)
- pixelTriplets+StripQuadruplets have a **lower fake (relative 20-40%)** than pixelTriplets
- We get **better resolution (relative 10-20%)** looking at CA reco tracks with **pT min of 0.55 GeV**



Additional plots here

TRacKing CA+Strips: 2023D scenario

- **Efficiency** compatible with pixelTriplets at high eta as well (within relative **5-10%**)
- pixelTriplets+StripQuadruplets have a **lower fake (relative 20-40%)** than pixelTriplets
- We get **better resolution (relative 10-20%)** looking at CA reco tracks with **pT min of 0.55 GeV**
- **Reconstruct less reco tracks due to lower fakes (relative >50%)**
-

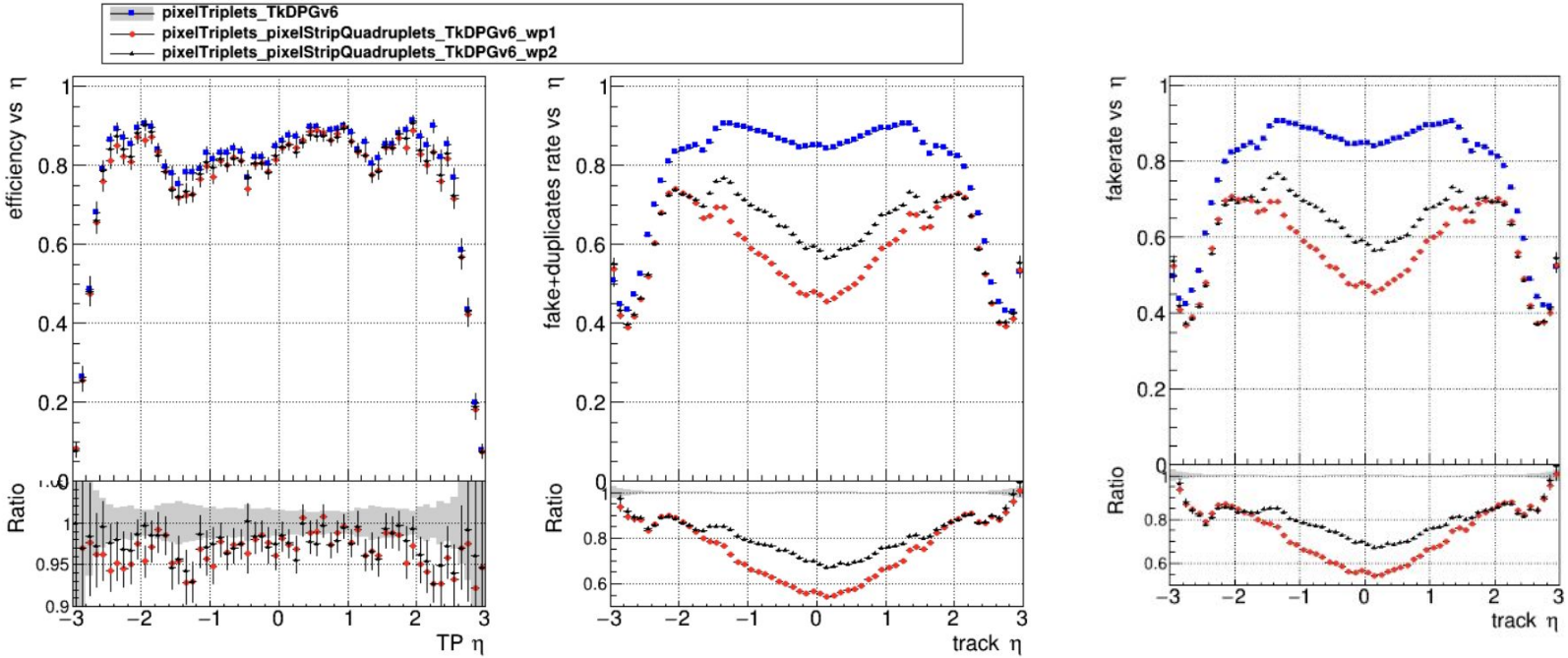


Additional plots here

TRacKing CA+Strips: TKDPGv6 scenario

GT: 140X_mcRun3_2024_realistic_EOR3_TkDPGv6
Dataset:
RelValTTbar_14TeV/GEN-SIM-DIGI-RAW/PU_140X_mcRun3_2024_realistic_EOR3_TkDPGv6_RV245_2024-v1/

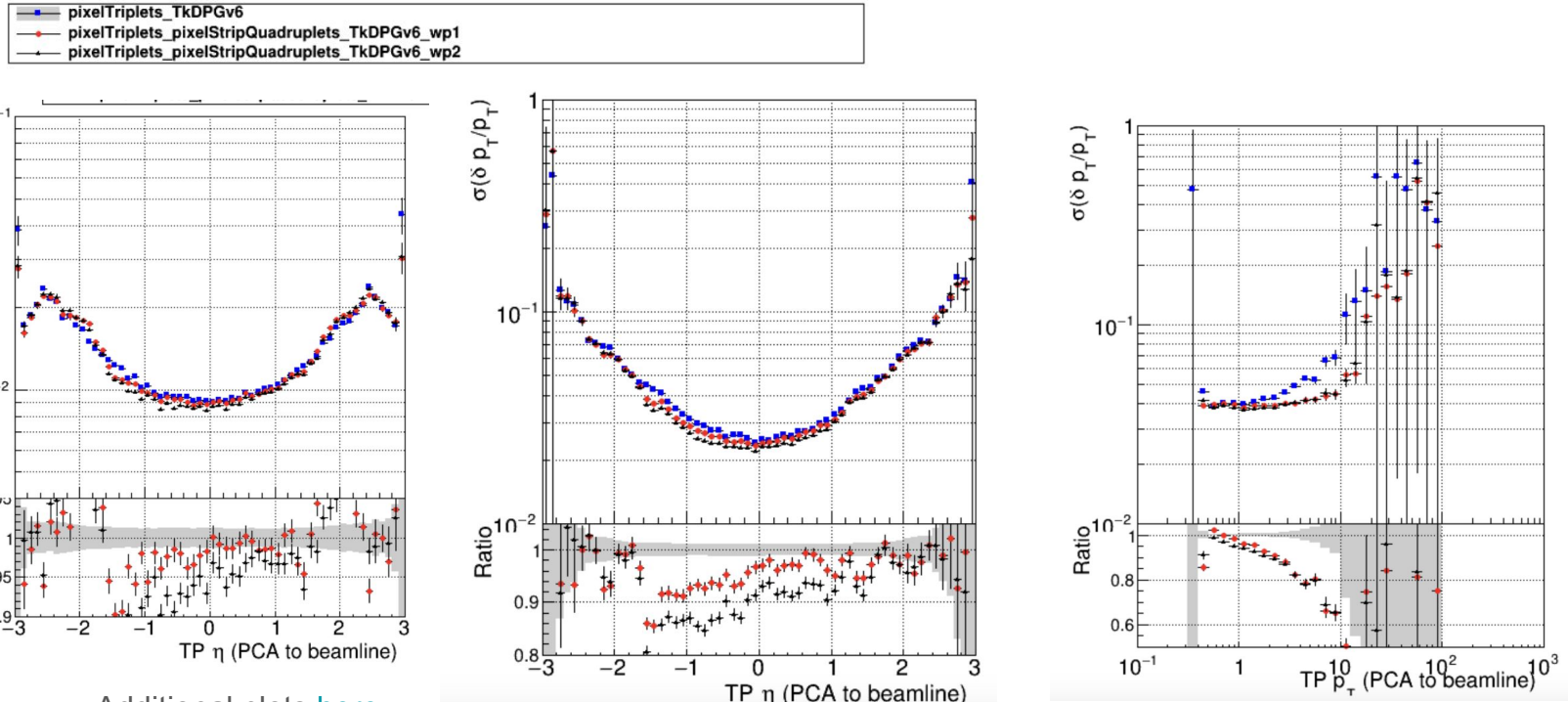
- **Efficiency** compatible with pixelTriplets at high eta as well,
- Overall slightly higher fakes as expected. pixelTriplets+StripQuadruplets have a **lower fake (relative 20-40%)** than pixelTriplets



Additional plots here

TRacKing CA+Strips: TKDPGv6 scenario

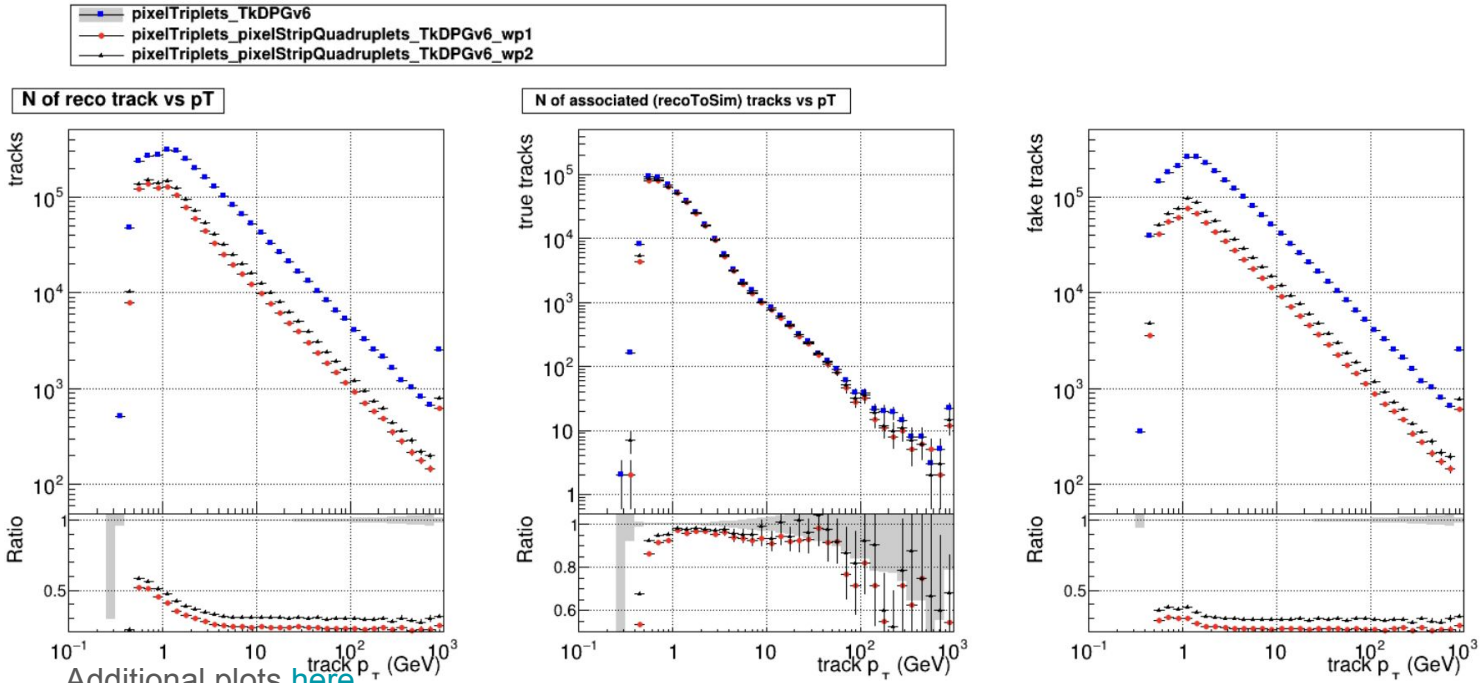
- **Efficiency** compatible with pixelTriplets at high eta as well,
- Overall slightly higher fakes as expected. pixelTriplets+StripQuadruplets have a **lower fake (relative 20-40%)** than pixelTriplets
- We get **better resolution (relative 10-20%)** looking at CA reco tracks with **pT min of 0.55 GeV**



Additional plots [here]

TRacKing CA+Strips: TKDPGv6 scenario

- **Efficiency** compatible with pixelTriplets at high eta as well,
- Overall slightly higher fakes as expected. pixelTriplets+StripQuadruplets have a **lower fake (relative 20-40%)** than pixelTriplets
- We get **better resolution (relative 10-20%)** looking at CA reco tracks with **pT min of 0.55 GeV**
- **Reconstruct less reco tracks due to lower fakes (relative >50%)**

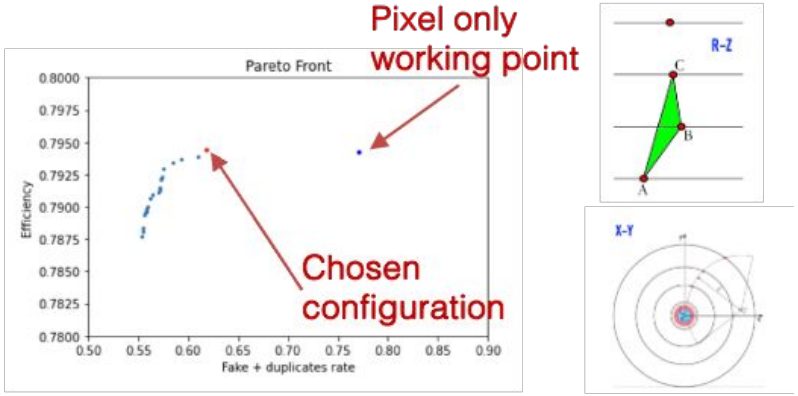


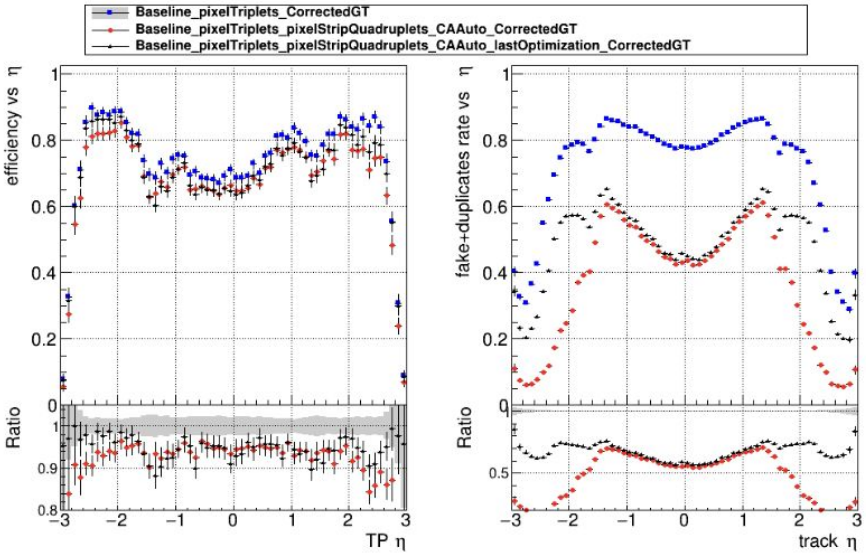
Additional plots here

TRacKing CA+Strips: Local Tracks @HLT

Particle Swarm Optimization (PSO) to optimize performance (**efficiency, fake+duplication rate**) by tuning CA parameters (e.g. azimuthal and spatial cuts for doublets and triplets) using the baseline for **End of Run 3 TT PU** (**worst scenarios for detector ageing**)

- Good performance also for HLT local reconstruction



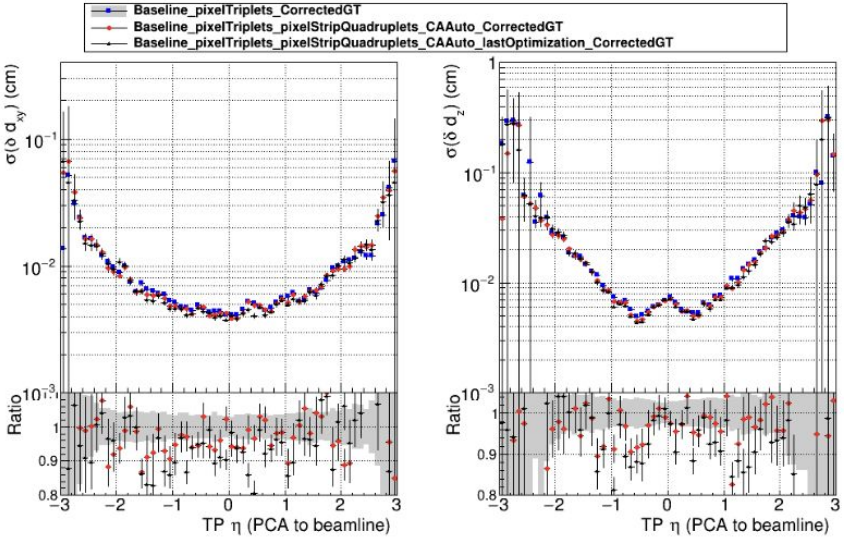


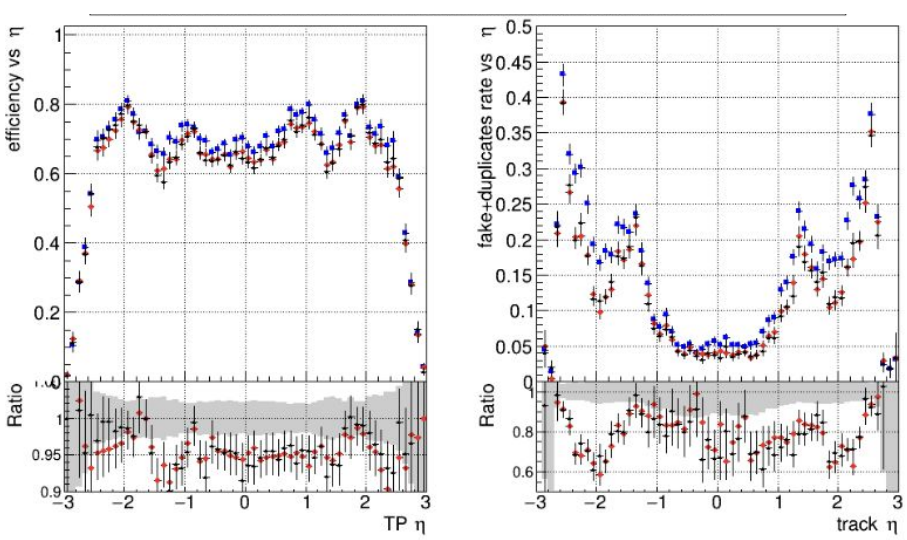
hltPixel tracks plots

TRacKing CA+Strips: Full Tracks @HLT

Implementation working both for offline and online reconstruction

- Similar performance (resolution) when looking at full tracks





Need to look at full tracks with no doublet recovery

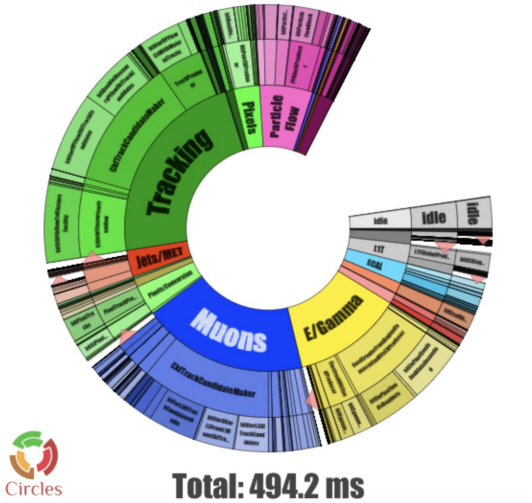
Full tracks plots

Global unpacking for strips @HLT:Timing Studies

https://twiki.cern.ch/twiki/bin/viewauth/CMS/TriggerStudiesTiming

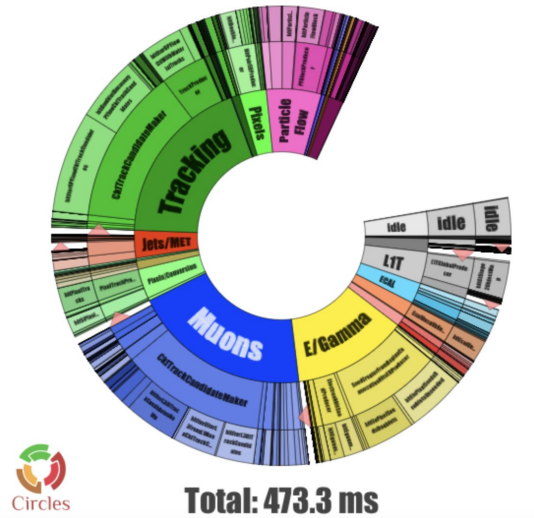
Latest release: **CMSSW_14_0_15_patch1** ,
Latest menu: **/dev/CMSSW_14_0_0/GRun/V182**

hltSiStripRawToClustersFacility.Demand=False
Global unpacking



Throughput
530.7 ± 4 evt/s

hltSiStripRawToClustersFacility.Demand=True
Default no global unpacking



Throughput
552.8 ± 2.8 evt/s

The default no global unpacking gives reduction of < 5% in throughput that can be compensated with CA speed up (to be measured!)

Lesson learned for Phase-2

- **Addition** of **tracker layers** at CA step (even using only the double-sided ones) should allow:
  - better resolution results
  - better fake rejections!

Ingredient needed missing once Run 3 is ok:

- Geometry structures to be read at runtime ready also for our customized structure (few months?)

```
namespace phase1PixelTopology {

  using pixelTopology::phi0p05;
  using pixelTopology::phi0p06;
  using pixelTopology::phi0p07;

  constexpr uint32_t numberOfLayers = 10;
  constexpr int nPairs = 13 + 2 + 4;; // without jump + jumping barrel + jumping forward
  constexpr uint16_t numberOfModules = 1856;

  constexpr uint32_t maxNumClustersPerModules = 1024;

  constexpr uint32_t max_ladder_bpx0 = 12;
  constexpr uint32_t first_ladder_bpx0 = 0;
  constexpr float module_length_bpx0 = 6.7f;
  constexpr float module_tolerance_bpx0 = 0.4f;  // projection to cylinder is inaccurate on BPIX1
  constexpr uint32_t max_ladder_bpx4 = 64;
  constexpr uint32_t first_ladder_bpx4 = 84;
  constexpr float radius_even_ladder = 15.815f;
  constexpr float radius_odd_ladder = 16.146f;
  constexpr float module_length_bpx4 = 6.7f;
  constexpr float module_tolerance_bpx4 = 0.2f;
  constexpr float barrel_z_length = 26.f;
```

CA doublets cuts

```
  HOST_DEVICE_CONSTANT uint8_t layerPairs[2 * nPairs] = {
      0, 1, 0, 4, 0, 7,           // BPIX1 (3)
      1, 2, 1, 4, 1, 7,           // BPIX2 (6)
      4, 5, 7, 8,                 // FPIX1 (8)
      2, 3, 2, 4, 2, 7, 5, 6, 8, 9,  // BPIX3 & FPIX2 (13)
      0, 2, 1, 3,                 // Jumping Barrel (15)
      0, 5, 0, 8,                 // Jumping Forward (BPIX1,FPIX2)
      4, 6, 7, 9                  // Jumping Forward (19)
  };

  HOST_DEVICE_CONSTANT int16_t phicuts[nPairs]{phi0p05,
                                               phi0p07,
                                               phi0p07,
                                               phi0p05,
                                               phi0p06,
                                               phi0p06,
                                               phi0p05,
                                               phi0p05,
                                               phi0p06,
                                               phi0p06,
                                               phi0p06,
                                               phi0p05,
                                               phi0p05,
                                               phi0p05,
                                               phi0p05,
                                               phi0p05,
                                               phi0p05,
                                               phi0p05,
                                               phi0p05,
                                               phi0p05,
                                               phi0p05};
  HOST_DEVICE_CONSTANT float minz[nPairs] = {
      -20., 0., -30., -22., 10., -30., -70., -70., -22., 15., -30, -70., -70., -20., -22., 0, -30., -70., -70.};
  HOST_DEVICE_CONSTANT float maxz[nPairs] = {
      20., 30., 0., 22., 30., -10., 70., 70., 22., 30., -15., 70., 70., 20., 22., 30., 0., 70., 70.};
  HOST_DEVICE_CONSTANT float maxr[nPairs] = {
      20., 9., 9., 20., 7., 7., 5., 5., 20., 6., 6., 5., 5., 20., 20., 9., 9., 9., 9.};
```

[SimplePixelTopology.h]

```
constexpr LayerPairData layerPairData[] = {
//  inner     outer       phicuts   minz    maxz  maxr
  { BPIX1,    BPIX2,      phi0p05,  -20.,    20.,  20.          }, //
  { BPIX1,    FPIX1Pos,   phi0p07,  0.,      30.,  9.           }, //
  { BPIX1,    FPIX1Neg,   phi0p07,  -30.,    0.,   9.           }, //
  { BPIX2,    BPIX3,      phi0p05,  -22.,    22.,  20.          }, //
  { BPIX2,    FPIX1Pos,   phi0p06,  10.,     30.,  7.           }, //
  { BPIX2,    FPIX1Neg,   phi0p06,  -30.,    -10., 7.           }, //
  { FPIX1Pos, FPIX2Pos,   phi0p05,  -70.,    70.,  5.           }, //
  { FPIX1Neg, FPIX2Neg,   phi0p05,  -70.,    70.,  5.           }, //
  { BPIX1,    BPIX3,      phi0p05,  -20.,    20.,  20.          }, //
  { BPIX2,    BPIX4,      phi0p05,  -22.,    22.,  20.          }, //
  { BPIX1,    FPIX2Pos,   phi0p05,  0.,      30.,  9.           }, //
  { BPIX1,    FPIX2Neg,   phi0p05,  -30.,    0.,   9.           }, //
  { FPIX1Pos, TIB1,       phi5deg,  -70.,    70.,  1000.        }, //
  { FPIX1Neg, TIB1,       phi5deg,  -70.,    70.,  1000.        }, //
  { BPIX3,    BPIX4,      phi0p06,  -22.,    22.,  20.          }, //
  { BPIX3,    FPIX1Pos,   phi0p06,  15.,     30.,  6.           }, //
  { BPIX3,    FPIX1Neg,   phi0p06,  -30,     -15., 6.           }, //
  { FPIX2Pos, FPIX3Pos,   phi0p05,  -70.,    70.,  5.           }, //
  { FPIX2Neg, FPIX3Neg,   phi0p05,  -70.,    70.,  5.           }, //
  { BPIX3,    TIB1,       phi5deg,  -22.,    22.,   1000.       }, //
  { BPIX4,    TIB1,       phi5deg,  -22.,    22.,   1000.       }, //
  { BPIX4,    TIB2,       phi5deg,  -22.,    22.,   1000.       }, //
  { TIB1,     TIB2,       phi5deg,  -55.,    55.,   1000.       }, //
  { FPIX2Pos, TIB1,       phi5deg,  -70.,    70.,   1000.       }, //
  { FPIX2Neg, TIB1,       phi5deg,  -70.,    70.,   1000.       }, //
  { FPIX3Pos, TIB1,       phi5deg,  -70.,    70.,   1000.       }, //
  { FPIX3Pos, TID1Pos2D,  phi5deg,  -70.,    70.,   1000.       }, //
  { FPIX3Pos, TID2Pos2D,  phi5deg,  -70.,    70.,   1000.       }, //
  { FPIX3Pos, TID3Pos2D,  phi5deg,  -70.,    70.,   1000.       }, //
  { FPIX3Neg, TIB1,       phi5deg,  -70.,    70.,   1000.       }, //
  { FPIX3Neg, TID1Neg2D,  phi5deg,  -70.,    70.,   1000.       }, //
  { FPIX3Neg, TID2Neg2D,  phi5deg,  -70.,    70.,   1000.       }, //
  { FPIX3Neg, TID3Neg2D,  phi5deg,  -70.,    70.,   1000.       }, //
  { TID1Pos2D,TID2Pos2D,  phi0p09,  -1000., 1000.,  1000.       }, //
  { TID2Pos2D,TID3Pos2D,  phi0p09,  -1000., 1000.,  1000.       }, //
  { TID1Neg2D,TID2Neg2D,  phi0p09,  -1000., 1000.,  1000.       }, //
  { TID2Neg2D,TID3Neg2D,  phi0p09,  -1000., 1000.,  1000.       }, //
  { TIB1,     TID1Pos2D,  phi5deg,  0.,      55.,   1000.       }, //
  { TIB1,     TID1Neg2D,  phi5deg,  -55.,    0.,    1000.       }, //
  { TIB2,     TID1Pos2D,  phi5deg,  0.,      55.,   1000.       }, //
  { TIB2,     TID1Neg2D,  phi5deg,  -55.,    0.,    1000.       }, //
  { BPIX2,    TIB1,       phi0p09,  -22.,    0.,    1000.       }, //
```

[SimplePixelStripTopology.h]

TRacKing CA+Strips: Work in progress and plans

Ongoing effort:

- Check (physics, timing) performance the menu (offline, not in confDB) in Alpaka without pixel doublet recovery

Further future goals:

- Check how we can split the OT layers to consider intermediate pairs.
  - Differently from the IT, the OT modules are much more “scattered” and we could use a more “fine” splitting w.r.t. the layer one we have now.
- Improving fit:
  - Take into account the material gaps (pixel-pixel, pixel-strip,strip-strip) for multiple scattering.