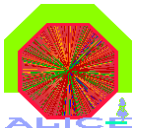


Status of ITS tracking

A. Dainese (INFN Legnaro)



News on ITS tracking, since last OW

- ◆ Material budget from TGeo (access mode optimized for speed and physics performance)
- ◆ Possibility to skip layers (AliITTrackerMI, AliITTrackerSA)
 - ✦ possible to do “tracking” with SPD alone (cosmics)
- ◆ Possibility to prolong TPC tracks to the SPD for $|\eta| > 1$
 - ✦ needs thorough testing with new ITS geometry (more realistic material description)
- ◆ Possibility to access the OCDB during tracking and check for a dead/noisy modules/chips
- ◆ Added in the AliESDtrack info on track's history in ITS
- ◆ Steering class AliITSRecoParam got fatter
 - ✦ next step, store it / retrieve it from OCDB
- ◆ Several options for cluster error parametrization in AliITSClusterParam
- ◆ Validation of tracking with the new ITS geometry (v11Hybrid)
 - ✦ OK now, after long debugging phase

Material Correction in AliTTrackerMI

- ◆ Material correction grouped in 4 methods:
 - ✦ CorrectForTPCtoITSMaterial (material from TGeo)
 - ✦ **CorrectForPipeMaterial (called 4 times)**
 - ✦ **CorrectForShieldMaterial (called 3 times)**
 - ✦ **CorrectForLayerMaterial (called 3 times)**
- ◆ For the last 3 methods (pipe, shields, layers), different ways to get the budget implemented.
- ◆ Default: **track-level TGeo look-up table during INWARD and OUTWARD, normal TGeo during REFIT**
 - ✦ optimal performance (resolutions and pulls) with tolerable (50%) increase of CPU time

Skipping Layers

- Why? Detector efficiency studies, layers badly aligned/calibrated
 - noisy SDD and SSD in cosmic runs

- Can specify in rec.C layers (up to 5) to be skipped:

```
AliITSRecoParam * itsRecoParam = AliITSRecoParam::GetLowFluxParam();
```

```
itsRecoParam->SetLayerToSkip(#);
```

```
AliITSReconstructor::SetRecoParam(itsRecoParam);
```

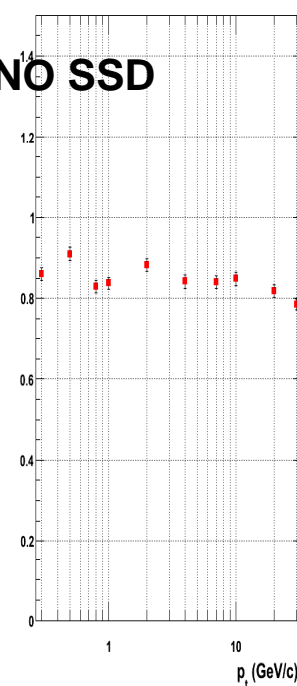
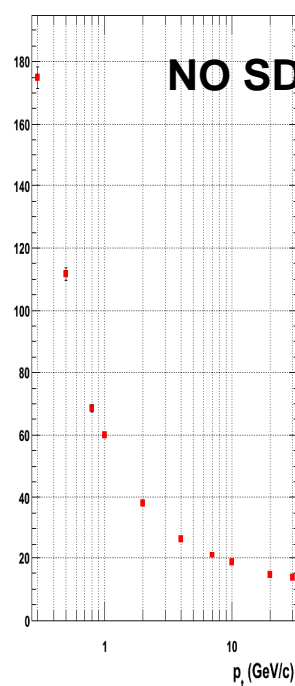
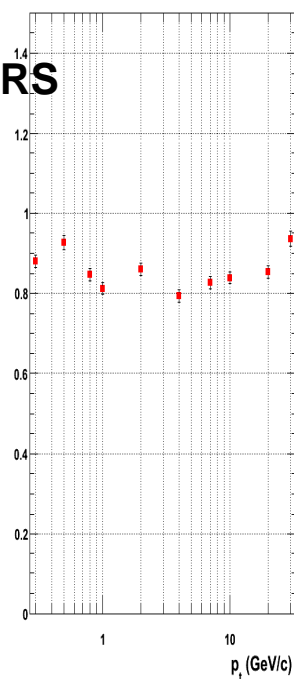
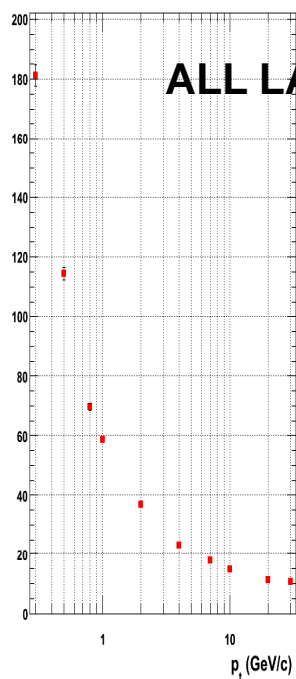
$d_0(r_0)$ Resolution for pions using True Vertex

$d_0(r_0)$ Pull for pions using True Vertex

d_0 res & pull

$d_0(r_0)$ Resolution for pions using True Vertex

$d_0(r_0)$ Pull for pions using True Vertex



ITS Tracking with Extended η Acc.

Motivation:

- ✦ TPC can reconstruct (with poorer quality) tracks in $0.9 < |\eta| < 1.4$
- ✦ These tracks are outside SSD acc. but inside (SDD) SPD acc.
- ✦ Assigning SPD points could allow to “do some physics” with these tracks (e.g. count primaries)

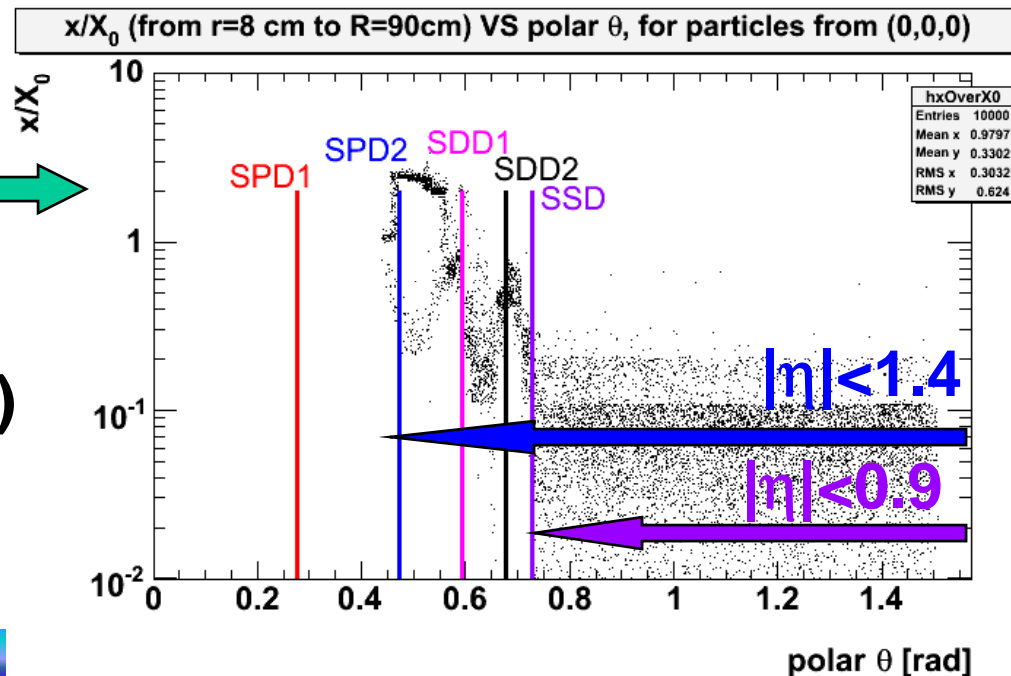
- ✦ **Warning:** large amount of material in $0.9 < |\eta| < 1.4$ (ITS supports, cables, electronics); only part of it in current ITS geometry
vPPRasymmFMD (more in v11)

vPPRasymmFMD:

x/X_0 vs polar θ



x/X_0 up to 200% (!!!)



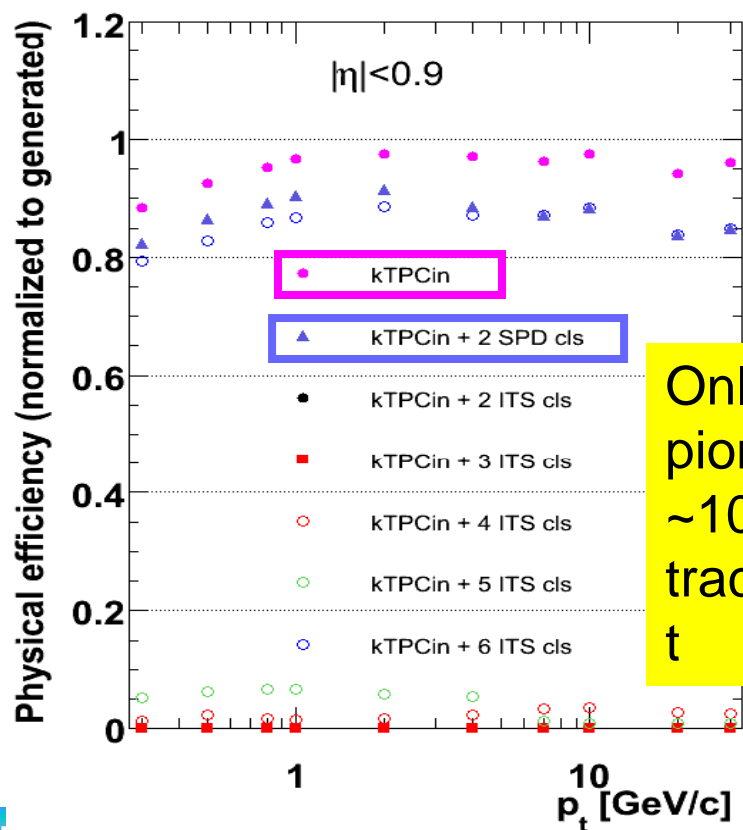
ITS Tracking with Extended η Acc.

Roadmap:

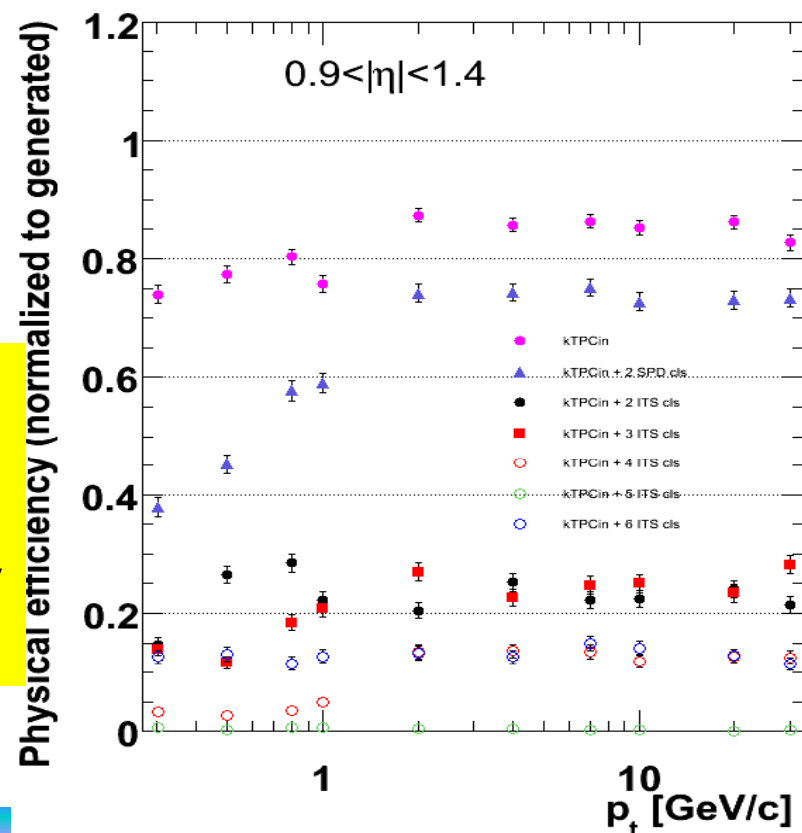
- ✦ Possibility to try to prolong those tracks in ITS (DONE)
- ✦ Careful study of efficiencies and track quality (to be done with new v11 ITS geometry)

- ✦ ExtendedEta option added (off by default) in AliITSRecoParam

Physical tracking efficiency (B=0.5T, no misal.)



Physical tracking efficiency (B=0.5T, no misal.)

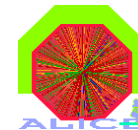


Class AliITSClusterParam

- ◆ The method `GetError` can return 3 types of errors, depending on the value of `AliITSRecoParam::GetClusterErrorsParam`:
 - ✦ 0: original error (as stored in `AliITSRecPoint`, i.e. nominal resol.)
 - ✦ 1: error param by Marian (original code from `AliITStrackerMI`)
 - ✦ 2: new error param with angular dependence, by Marcello
- ◆ Little effect on tracking performance, but Millepede realigns better with parametrization #2
- ◆ Next: possibility to add an error to account for misalignment / miscalibration (will be needed for high- p_t track finding with B on)



Treatment of bad chips/detectors development in progress



- ◆ **BAD SINGLE SPD PIXELS AND SDD ANODES:**

ClusterFinders read them from the DB and skip them in cluster creation.

Tracker, in constructor, reads these bad from DB and skips the layer if there is only a bad in the search road.

- ◆ **BAD SSD STRIPS:**

ClusterFinder creates fake clusters with $Q=0$.

Tracker skips the layer if there is only a $Q=0$ cluster in the search road.

- ◆ **BAD CHIPS & MODULES:**

Tracker reads from the DB the list of dead modules and of bad chips per module. This is done only once in the constructor of AliITStrackerMI and the info is stored in its data members of the kind AliITSdetector (Bool_t fIsBad, Int_t fNChips, Bool_t fChipsIsBad[fNChips]).

AliITSsegmentation changed to provide a method that takes (module #, xlocal, zlocal) and returns chip #.

AliITSCalibration changed to provide a method that gives fNChips and the array fChipsIsBad.

Tracker skips the layer if there is only a bad module/chip in the search road.

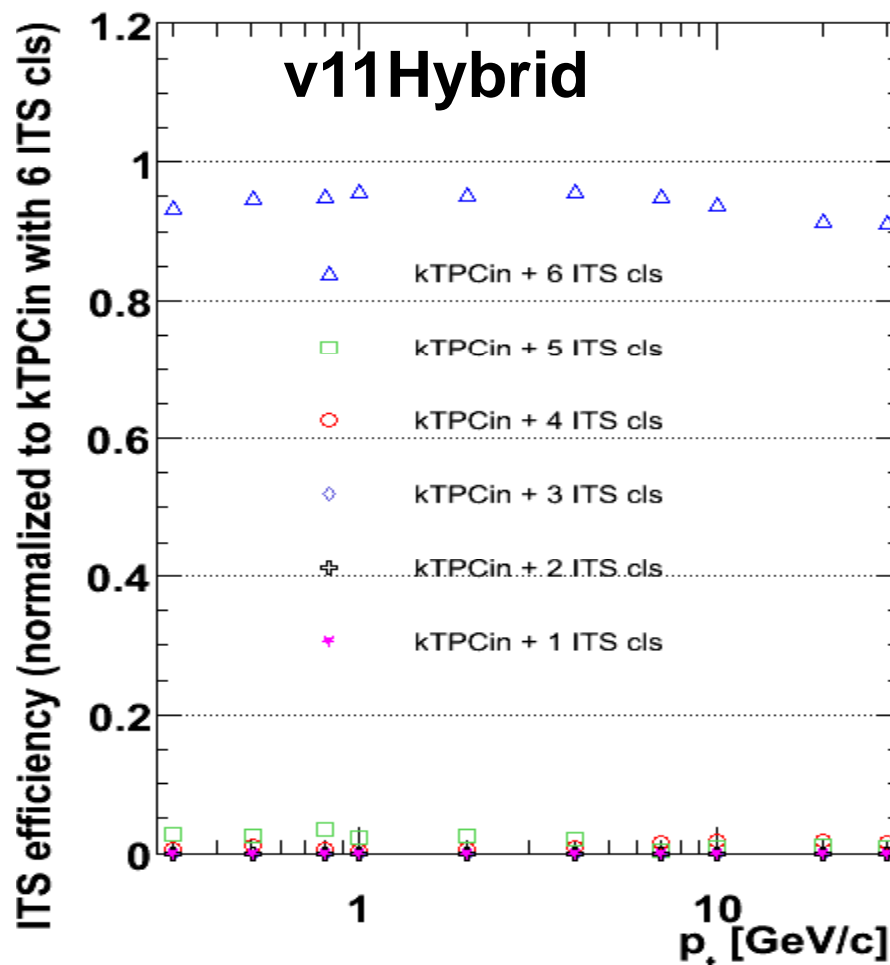
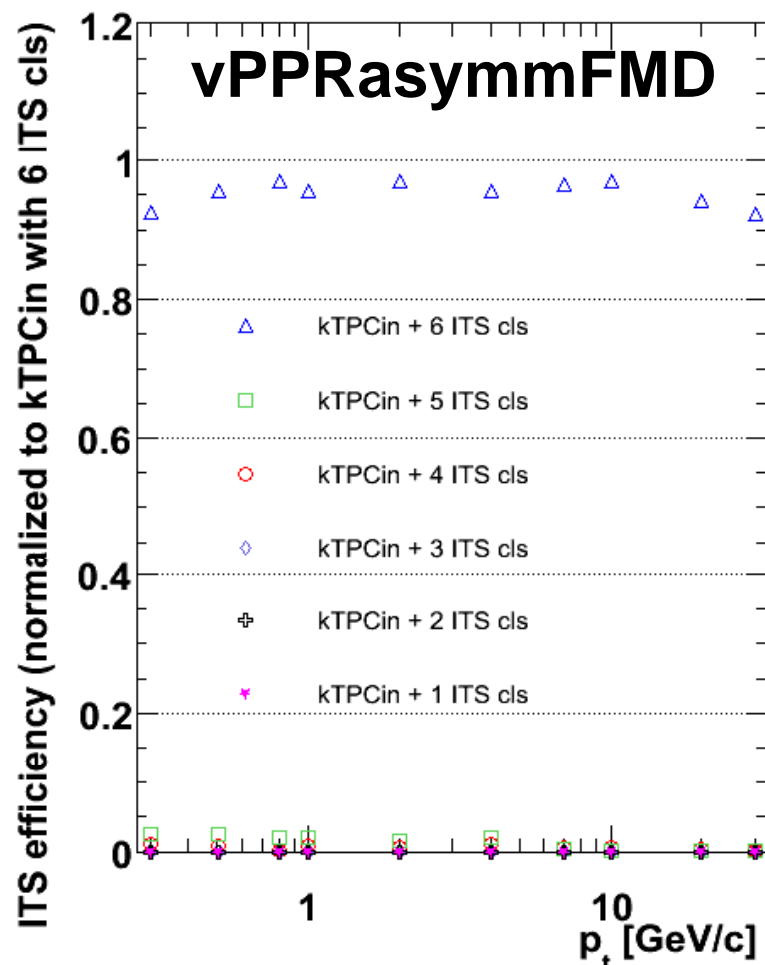
History of AliESDtrack in ITS

- ◆ Original proposal (Cvetan): store to the AliESDtrack the # of module crossed by the tracks in each ITS layer
- ◆ Motivation:
 - ✦ study/check of the ITS det. efficiencies
 - ✦ debugging/understanding of tracking efficiencies
 - ✦ track selection criteria
- ◆ Implementation:
 - ✦ added a `Int_t fITSMModule[12]` data member to AliESDtrack
 - ✦ filled in AliITStrackerMI::Clusters2Tracks (ITS inward) & also in AliITStrackerMI::RefitInward (ITS refit → both MI and SA)
- ◆ Method AliESDtrack::GetITSMODULEIndexInfo(layer, status, module, xloc, zloc)
 - ✦ **module**: module # (numbered as in the CDB)
 - ✦ **xloc , zloc** : crossing point of track on module (local module ref. frame), in cm with precision of 1 mm
 - ✦ **status**: **1** "found" (cluster is associated), **2** "dead" (module is dead from OCDB), **3** "skipped" (module or layer forced to be skipped), **4** "outinz" (track out of z acceptance), **5** "nocls" (no clusters in the road), **6** "norefit" (cluster rejected during refit), **7** "deadzspd" (holes in z in SPD)

Validation of tracking with v11Hybrid

ITS tracking efficiency (B=0.5T, no misal.)

ITS tracking efficiency (B=0.5T, no misal.)

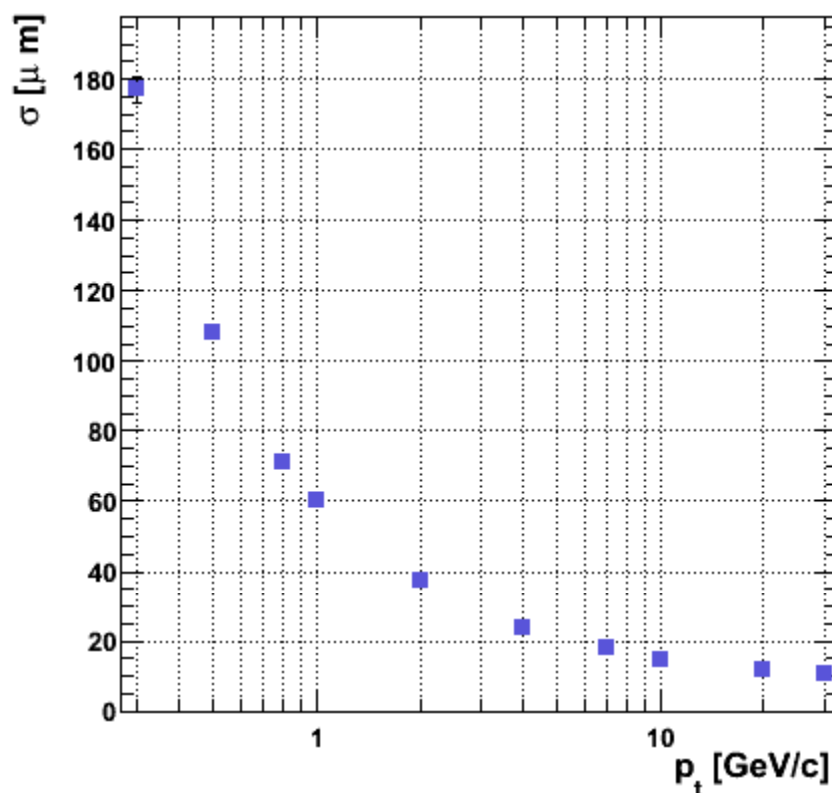


OK, after several bug fixes

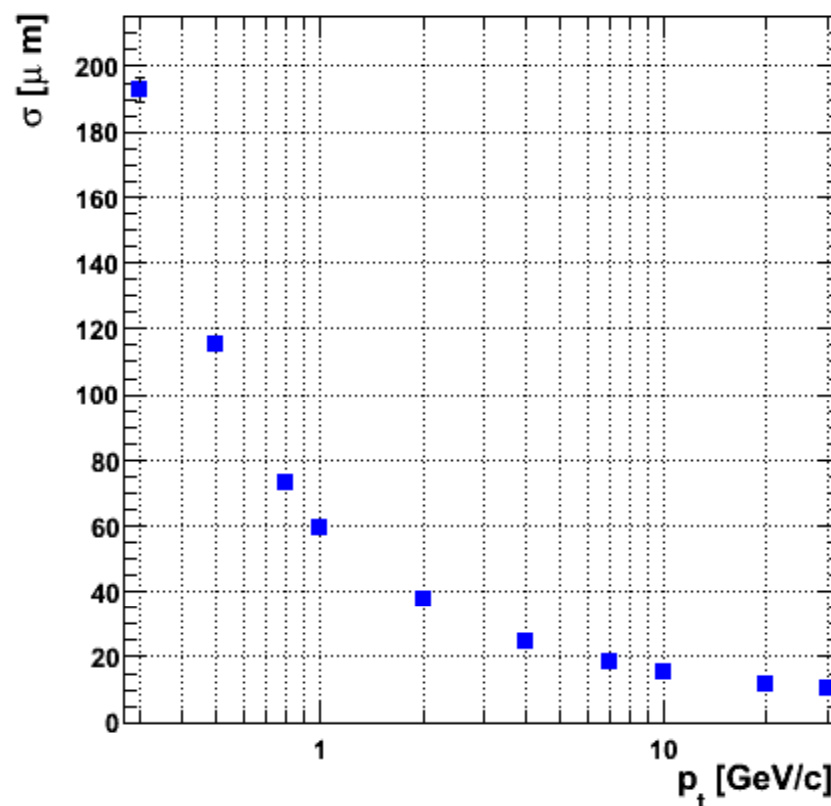
vPPRasymmFMD

v11Hybrid

$d_0(r\phi)$ Resolution using True Vertex

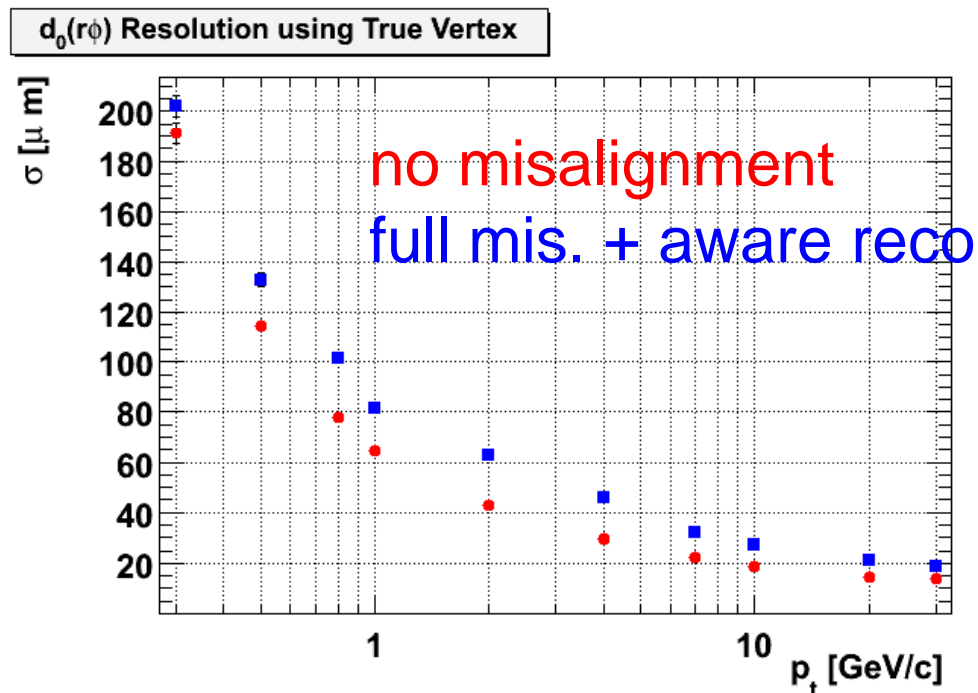


$d_0(r\phi)$ Resolution using True Vertex



OK, after several bug fixes

- ◆ Alignment-aware reco with large misalignments



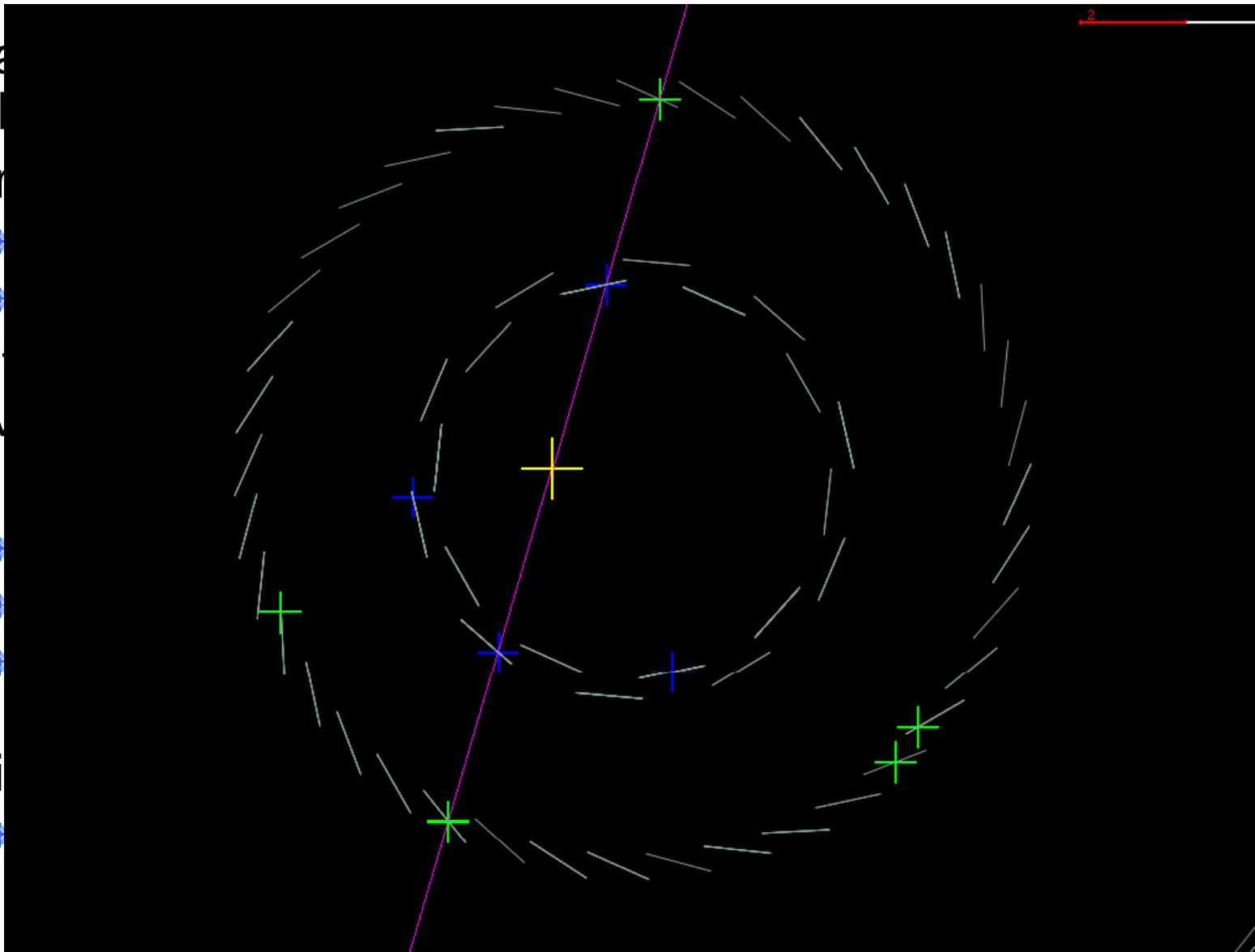
also with old geom

→ needs to be solved

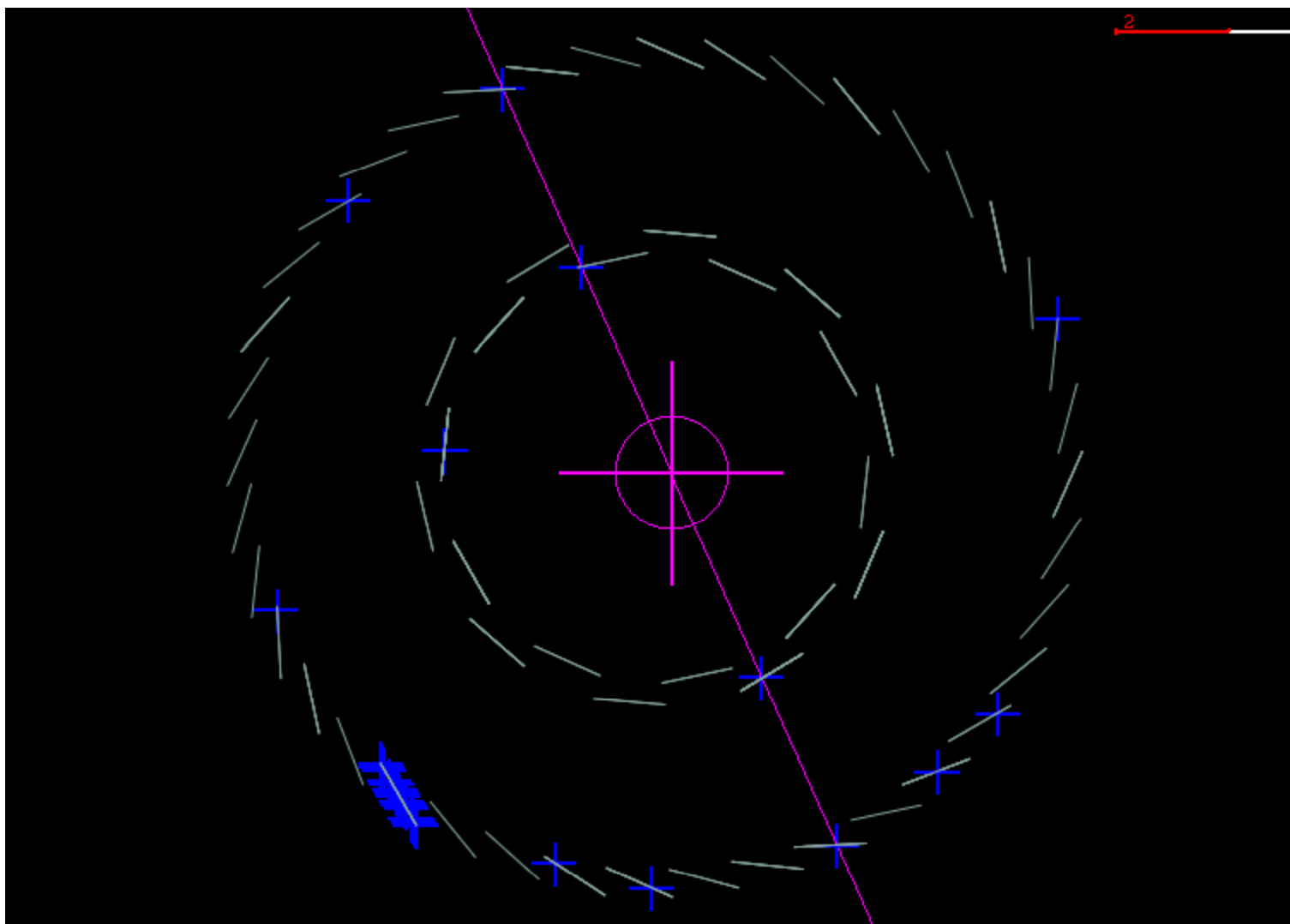
- ◆ AliITSRecoParam to OCDB
- ◆ QA of ITS tracking

Cosmics tracking

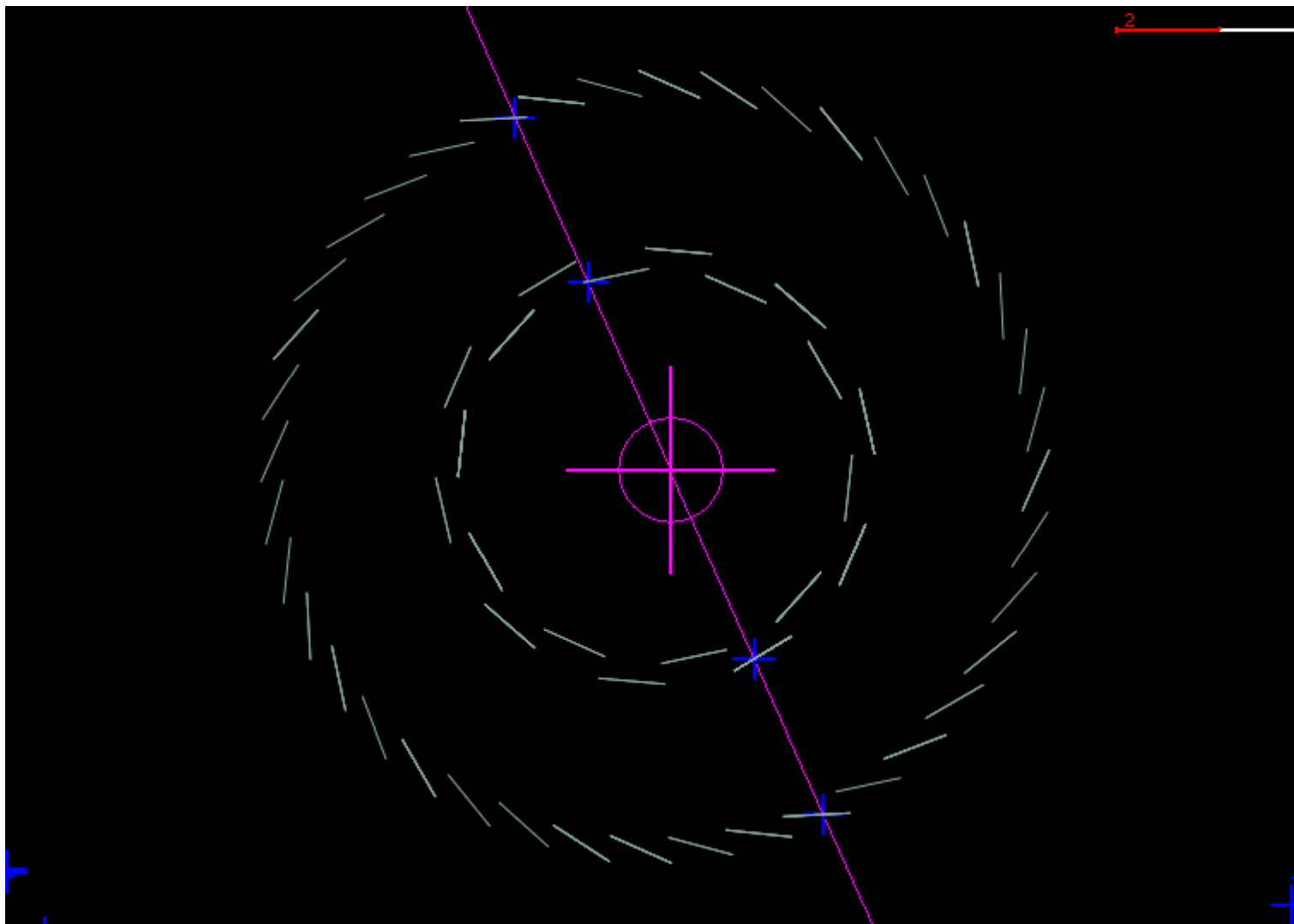
- ◆ Fast
- ◆ Simple
- ◆ Track
- ◆ Over
- ◆ Over
- ◆ Fast



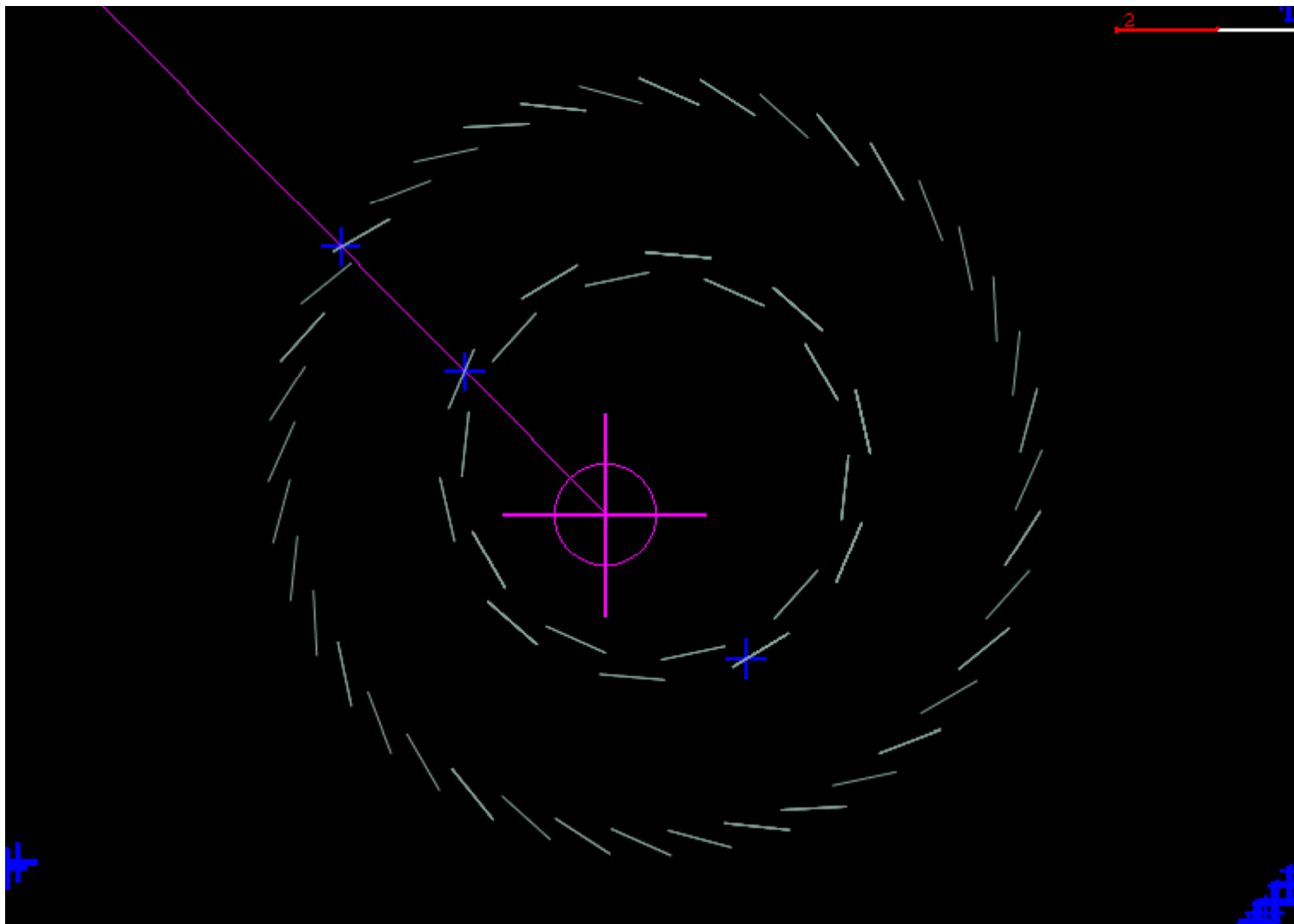
Cosmics tracking: Optimization



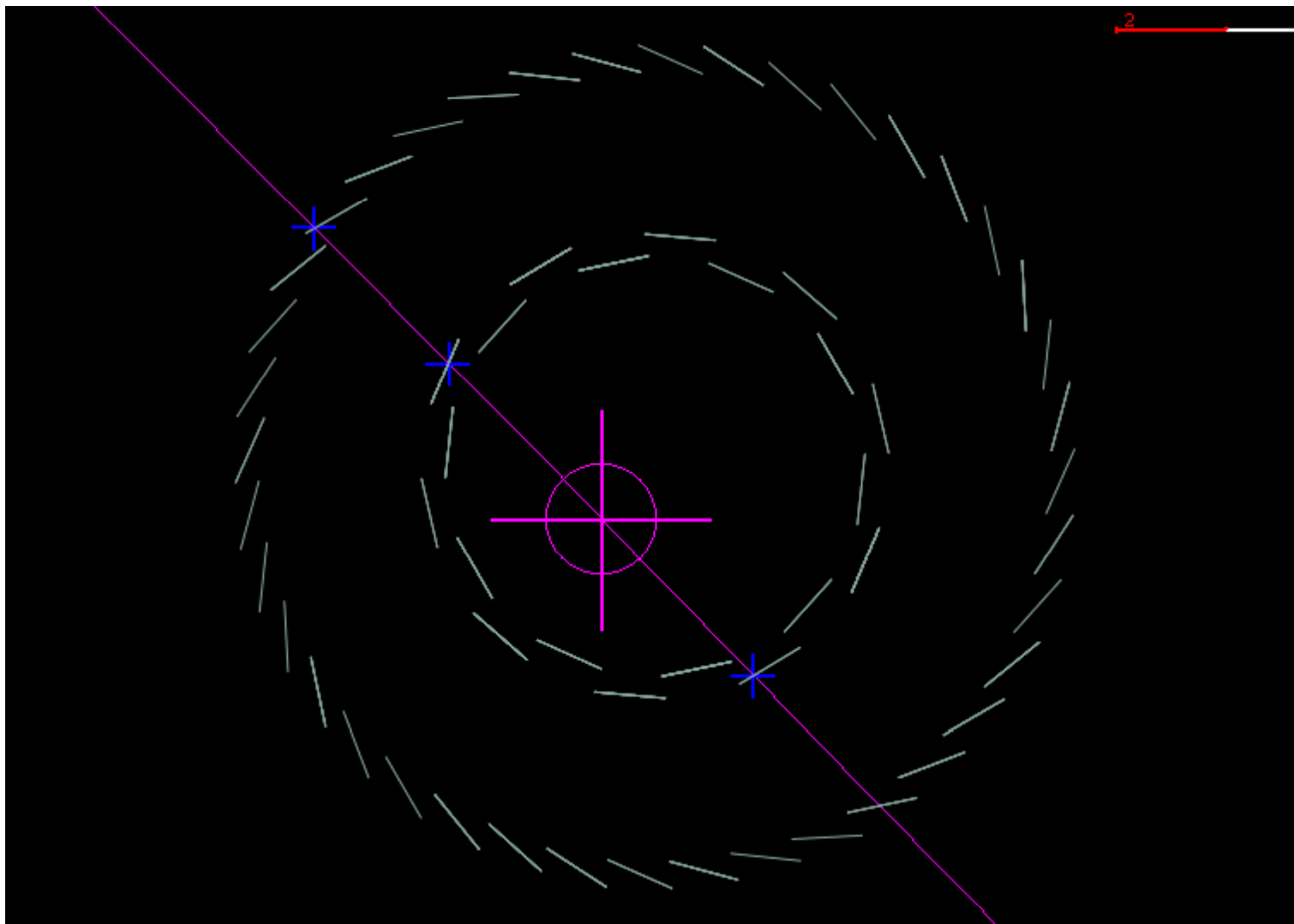
Cosmics tracking: Optimization



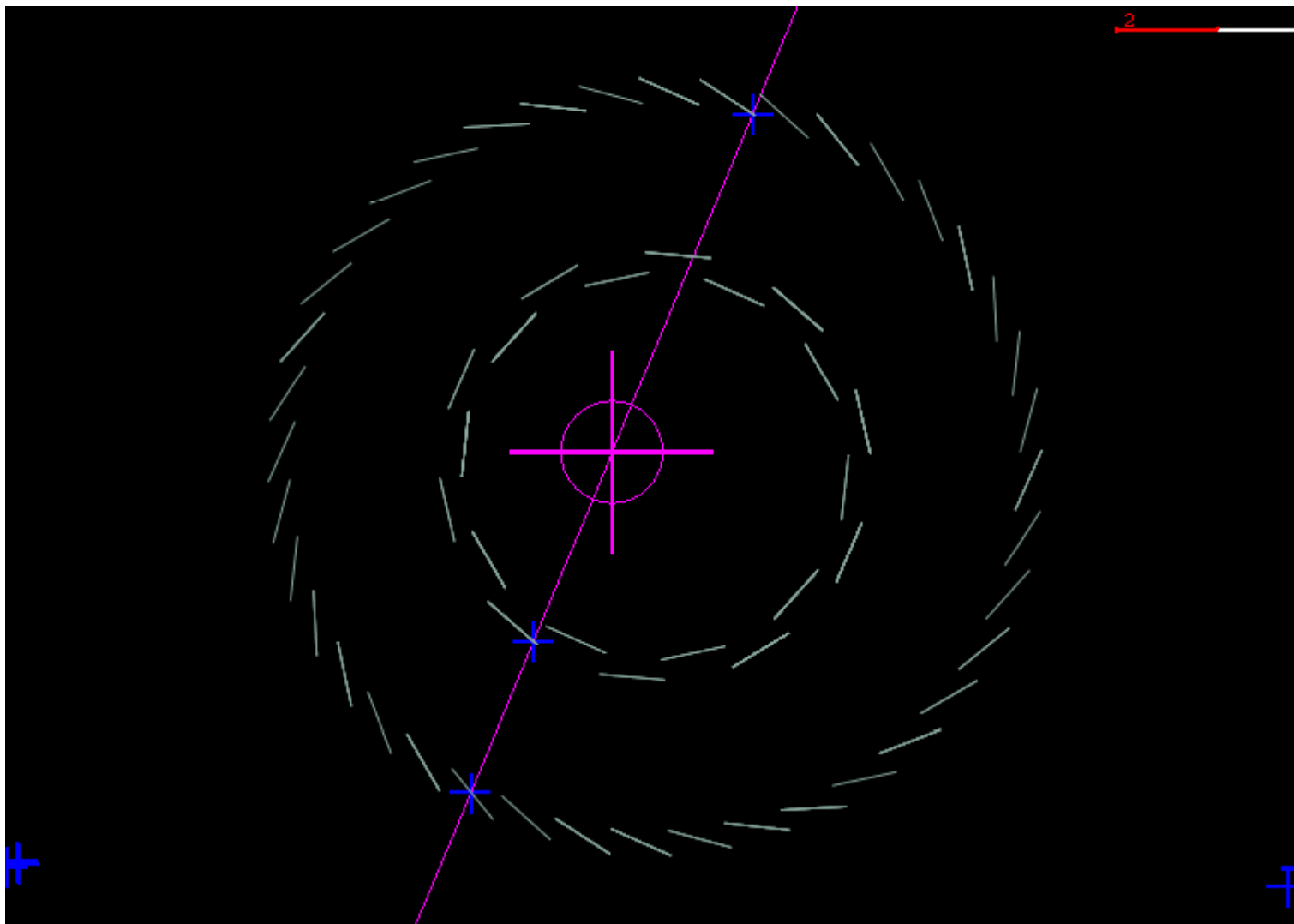
Cosmics tracking: Optimization



Cosmics tracking: Optimization

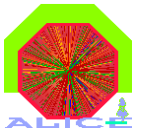


Cosmics tracking: Optimization





Cosmics tracking: plans for next run



- ◆ Commit changes from optimization done on this run
- ◆ Account for miscalibration and misalignment in cluster errors (via AliITSClusterParam)
 - ✦ needed for tracking with B on & for TPC-ITS matching
- ◆ Reconstruct and store ITS tracks twice:
 - ✦ prolongation of TPC tracks (AliITStrackerMI)
 - ✦ tracking in ITS stand-alone (AliITStrackerSA), needed for alignment
- ◆ Strategy? Duplicate tracks in one AliESDs.root or have two rec.C and two files (AliESDs.root & AliESD_ITSSa.root)?