Update on secondary track finder

Lia Lavezzi
INFN Pavia

EVO, Pattern Recognition Meeting,
January 22, 2013

Summary

- Structure to integrate primary and secondary track finder
- New implementation for secondary track finder
- New tool for STT

Prim. & Sec. PR Integration

I. Structure to integrate primary and secondary track finder

THE IDEA

❖ The **primary** track finder in tracking/ uses:
- MVD, STT, SciTil hits (so far)
- Conformal Transformation
- Legendre Transformation (for fitting)

❖ The **secondary** track finder (in the new version I am trying) uses:
- MVD, STT hits (so far)
- Conformal Transformation
- Legendre Transformation

they must be integrated

there must be a structure

Motivations

❖ In the primary track finder, the information is contained in several vectors and matrices → the idea is to create objects to contain such information and simplify the code.

❖ Though the FairHit already contains some information, some are missing

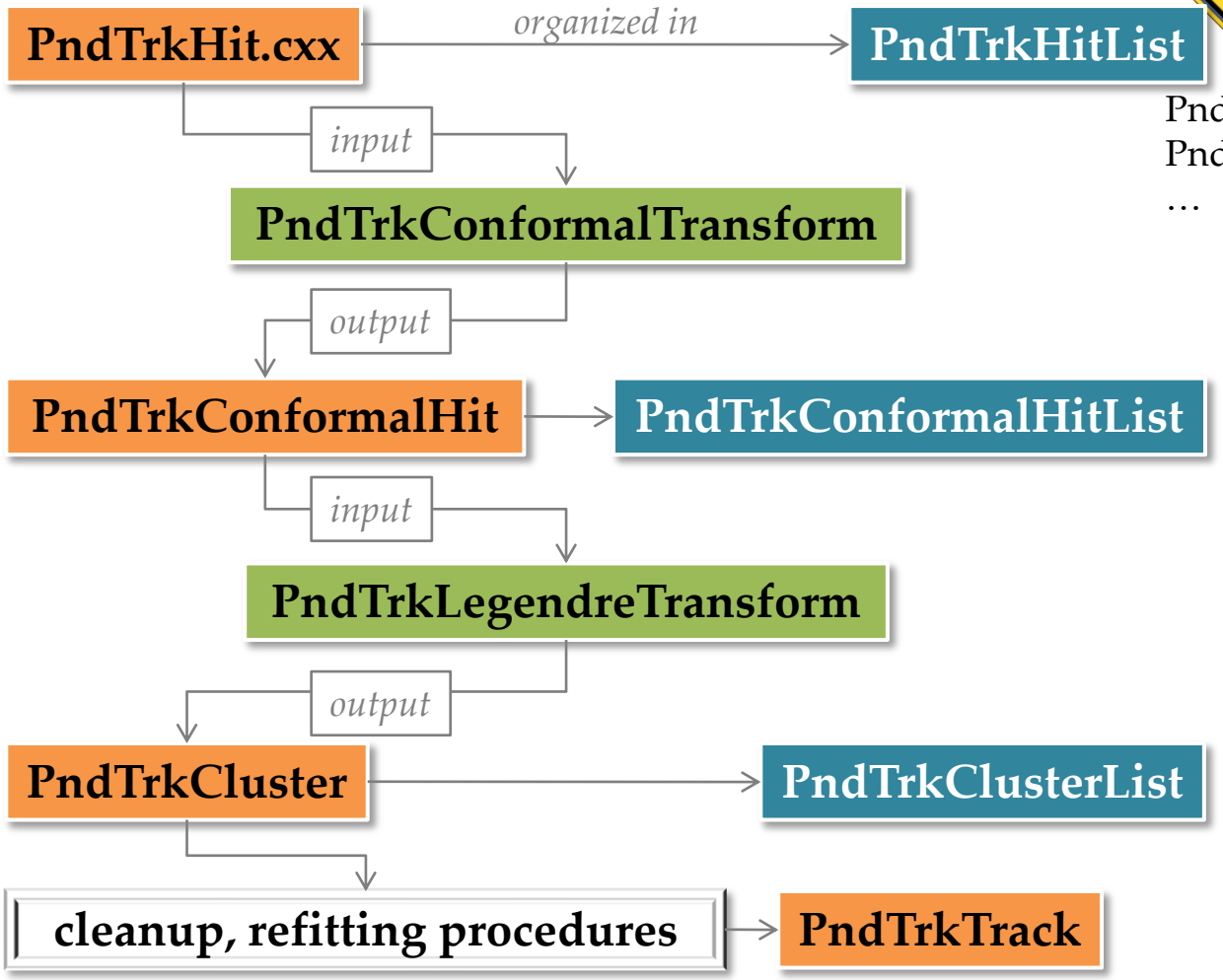
❖ This new structure resembles somehow the PndRecoHit of genfit: one hit type for all the detectors there, one PndTrkHit for all the detectors here.

❖ Concerning the output, the need to use the helix parameters, instead of the FairTrackPar at 1° and last hits, comes from the fact that they are easier to handle within the track finding code.

Structure

PndTrkLegendreTask

- TASK
- OBJECT
- LISTS
- TRANSFORM



https://subversion.gsi.de/fairroot/pandaroot/development/lia/tracking/secondary

PndTrkHit(List)

The TCAs of FairHit are transformed in PndTrkHitLists of PndTrkHit in order to handle all the hits with the same type of object

```
PndTrkHit(Int_t hitID, Int_t detID,
        Bool_t used,
        Int_t iregion,
        Int_t sensorID,
        TVector3& pos,
        Double_t isochrone,
        Double_t sortvar)
```

```
// IREGIONS ..........
// MVD
#define MVDPIXEL   0
#define MVDSTRIP   1
// STT
#define INNER_LEFT 3
#define SKEW_LEFT  5
#define OUTER_LEFT 7
#define INNER_RIGHT 2
#define SKEW_RIGHT  4
#define OUTER_RIGHT 6
            …
```

PndTrkHitLists

```
std::vector<PndTrkHit> hitlist;
```

These objects could replace the vectors that now are present in the primary track finder, because they contain all the information of the FairHit + some additional info

PndTrkCluster/Track

The cluster is just a vector of pointers to PndTrkHit.
It can be sorted from a certain 3D point (not yet implemented)
fIRegion depends on the regions of the PndTrkHit.

PndTrkCluster:

```
TVector3 fFromPoint;
Int_t fIRegion;
std::vector< PndTrkHit *> hitlist;
```

Once it is fitted, with the helix parameters, the PndTrkTrack is created:

PndTrkTrack:

```
Double_t fRadius, fCenterX, fCenterY,
       fTanL, fZ0;
Int_t fCharge;
PndTrkCluster *fCluster;
```

At the very end, the PndTrkCluster is translated to the PndTrackCand and the PndTrkTrack to the PndTrack

PndTrkConformalTransform

- It creates the PndTrkConformalHit
- It can transform both:
  - points (x, y)

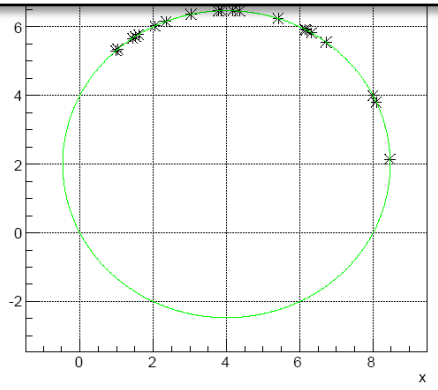
$$u = \frac{x}{x^2 + y^2}, \qquad v = \frac{y}{x^2 + y^2},$$

  - and circles (x, y, r_d)

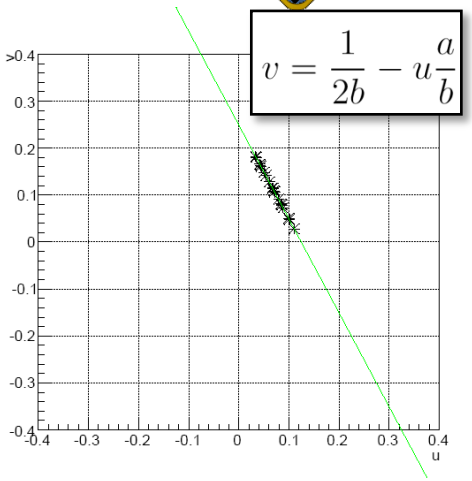
$$u = \frac{x}{x^2 + y^2 - r_d^2}, \qquad v = \frac{y}{x^2 + y^2 - r_d^2},$$

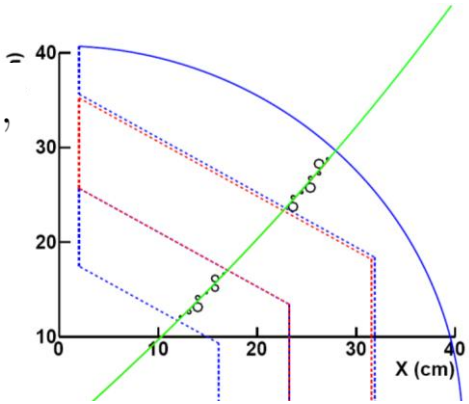
$$r_c = \frac{r_d}{x^2 + y^2 - r_d^2}$$

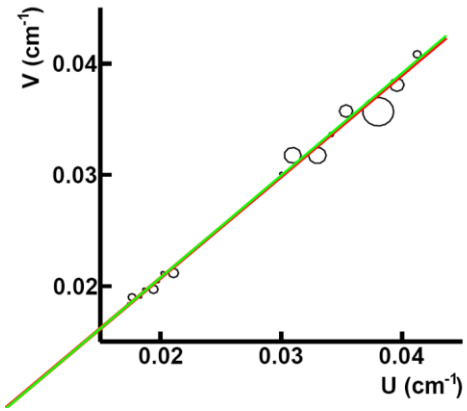
$$(x-a)^2 + (y-b)^2 = r^2 = a^2 + b^2$$



$$v = \frac{1}{2b} - u\frac{a}{b}$$



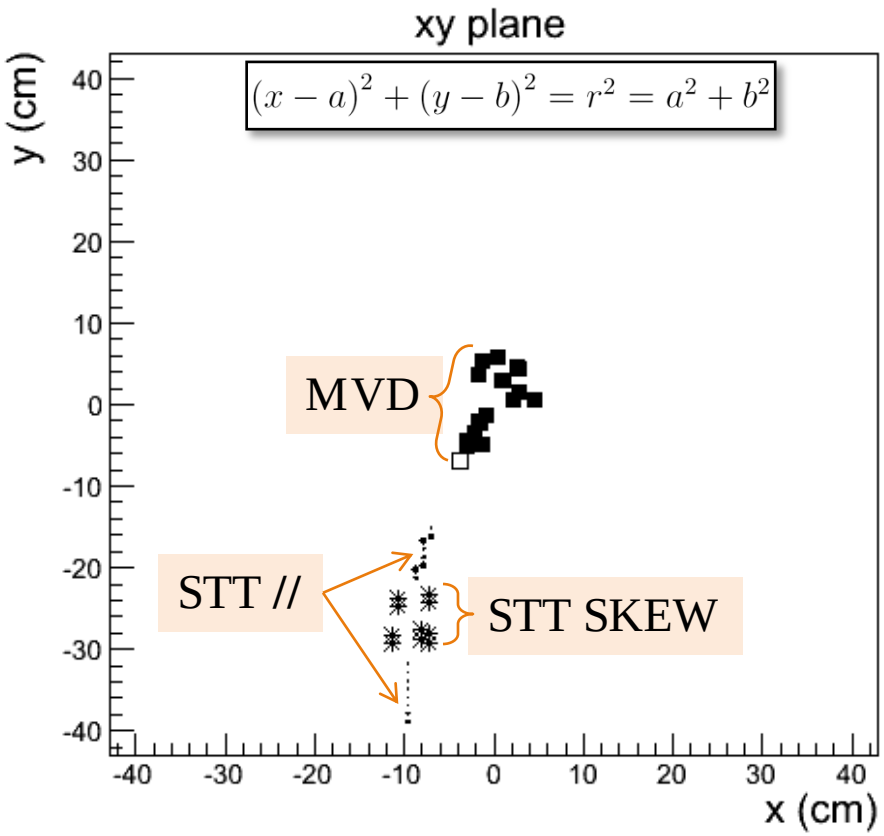


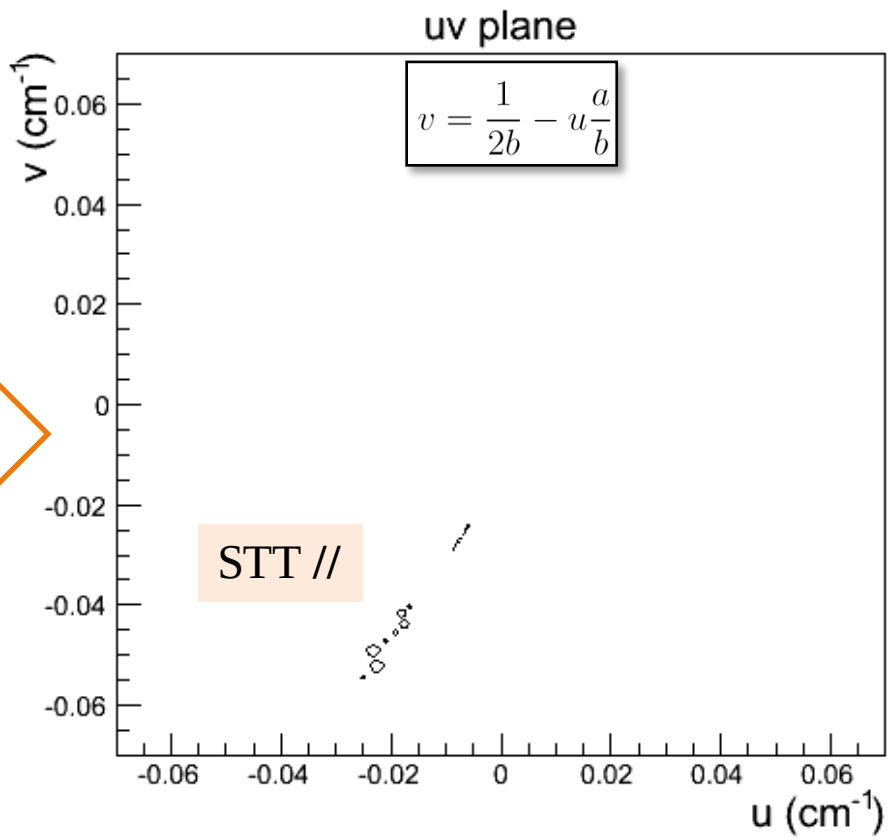


… to a conformal frame, centerd in any (X, Y)

PndTrkConformalTransform

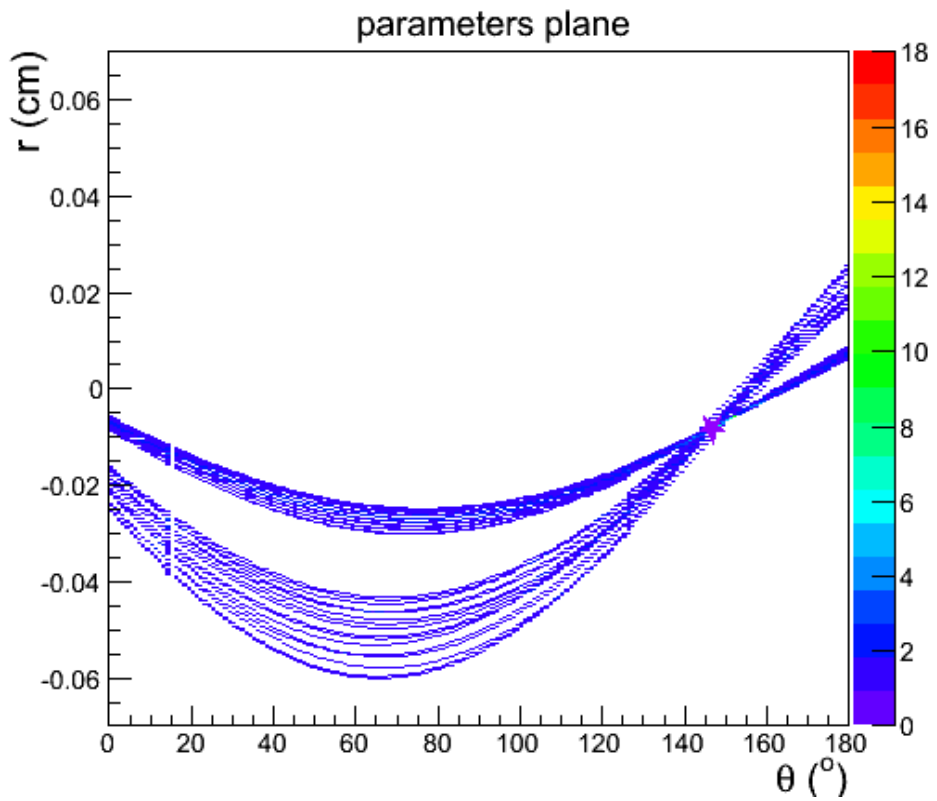
- It creates the PndTrkConformalHit

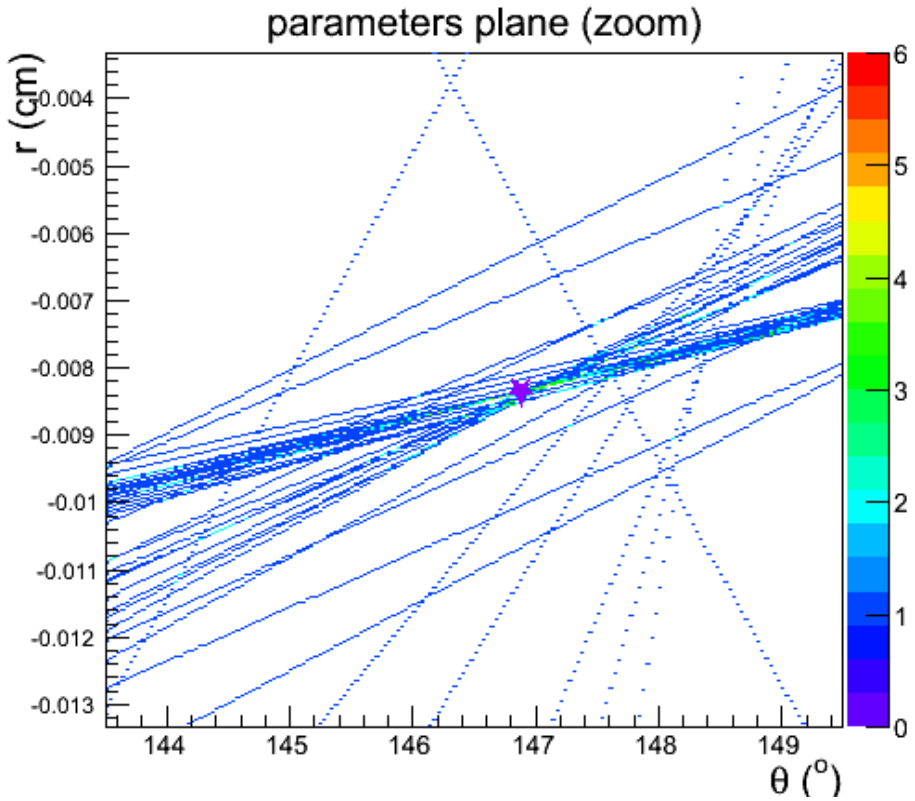




PndTrkLegendreTransform

- It fills a TH2F (for now) to use TH2F::GetMaximum() as peak finder.
- The parameter frame is





Procedure

II. New implementation for secondary track finder

- Up to now the secondary track finder was a sort of adaptation of the primary track finder to tracks not originating from (*0, 0, 0*).

- the impossibility to apply the constraint of being primary creates some problems, mainly:
  - the conformal map transforms circles in straight lines only if the circles pass in (0, 0)
  - the fit works fine only with a big lever arm, so without the (*0, 0*) point it might give underestimated radius circles (with GLPK)

- new implementation:
  - make a translation of the center onto a precise hit, such as an MVD hit, a SciTil hit or the center of the STT in case of small isochrone
  - apply the Legendre transformation and search for the peak in the accumulation histogram

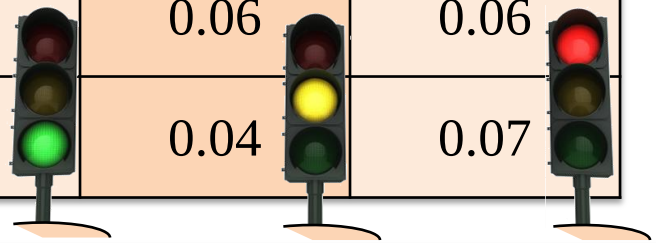
PndTrkLegendreTransform

Preliminary Results ***with 1000 muon track events from IP***

Reconstructable MC Track (> 3 STT // hits)

- *Reconstructed*
  - **Good** (>80% of right hits assigned)
  - **Wrong** (< 80% of right hits assigned)
- *Not Reconstructed* (**Missed**)

# Tracks	p (GeV/c)	Good	Wrong	Missed
1	0.3	0.91	0.07	0.02
4	0.3	0.94	0.03	0.04
1	1	0.88	0.06	0.06
4	1	0.88	0.04	0.07

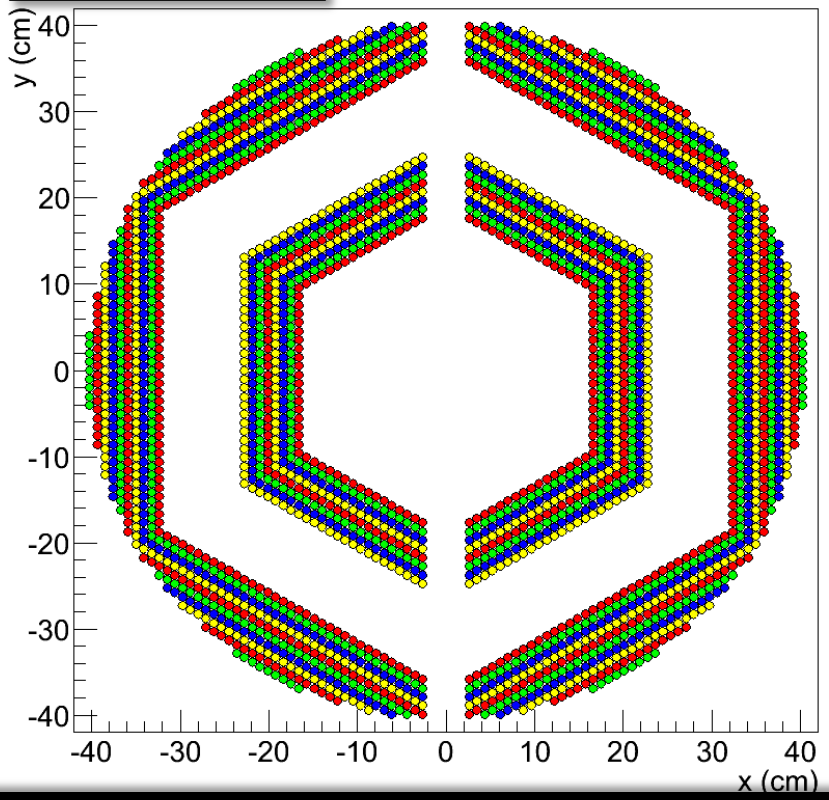


New functions for STT

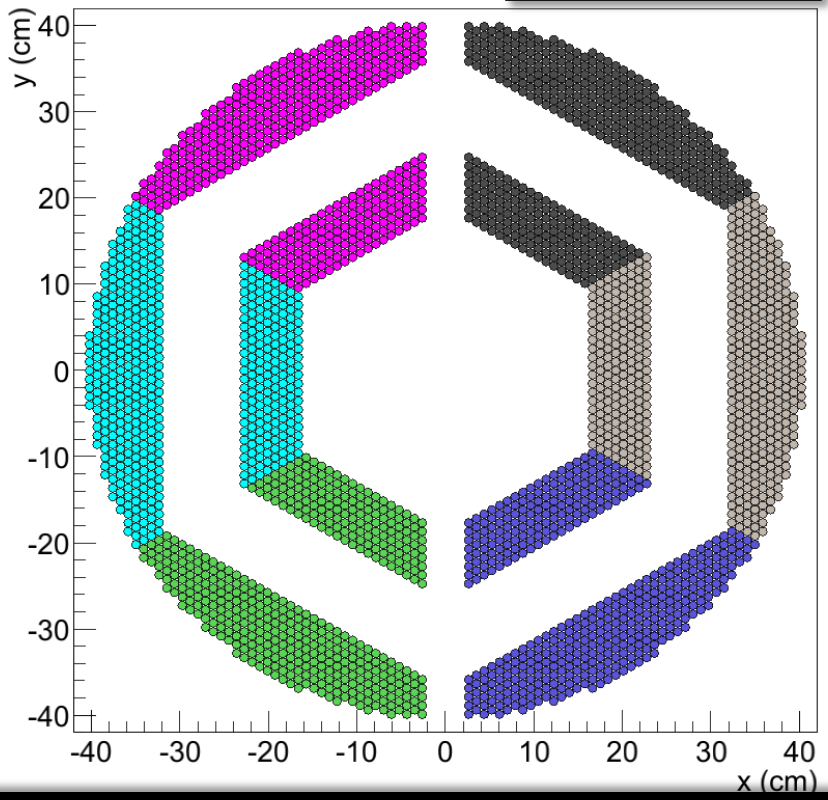
Parametrization of the geometry limiting tubes…
… in order to assign to each tube a layerID and a sectorID

III. Tool for STT

LAYERS

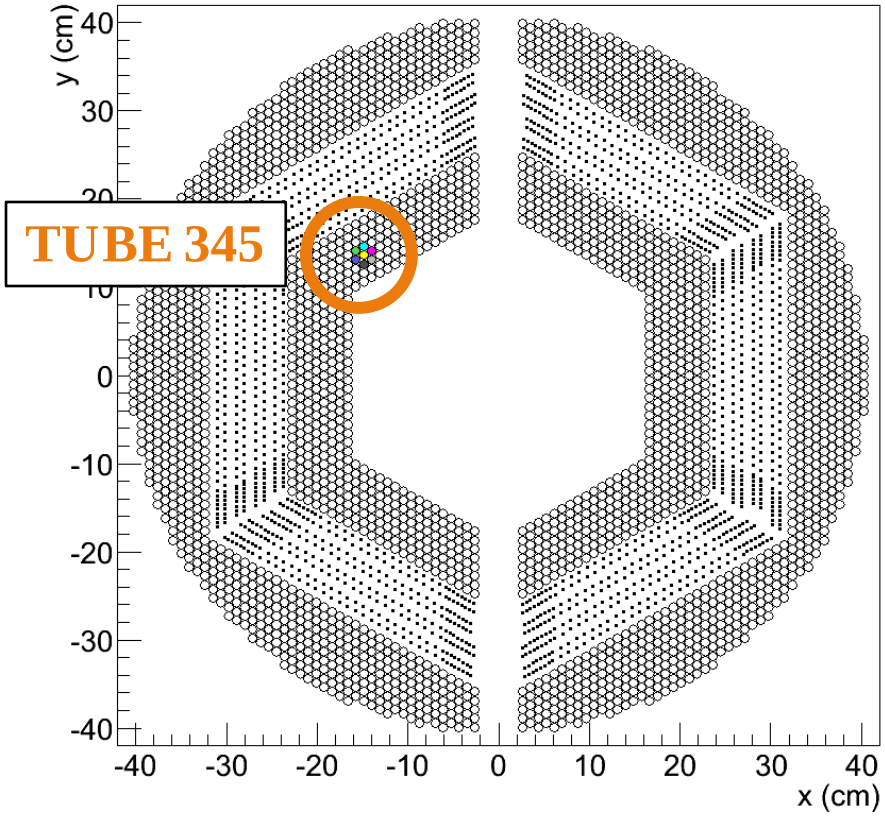


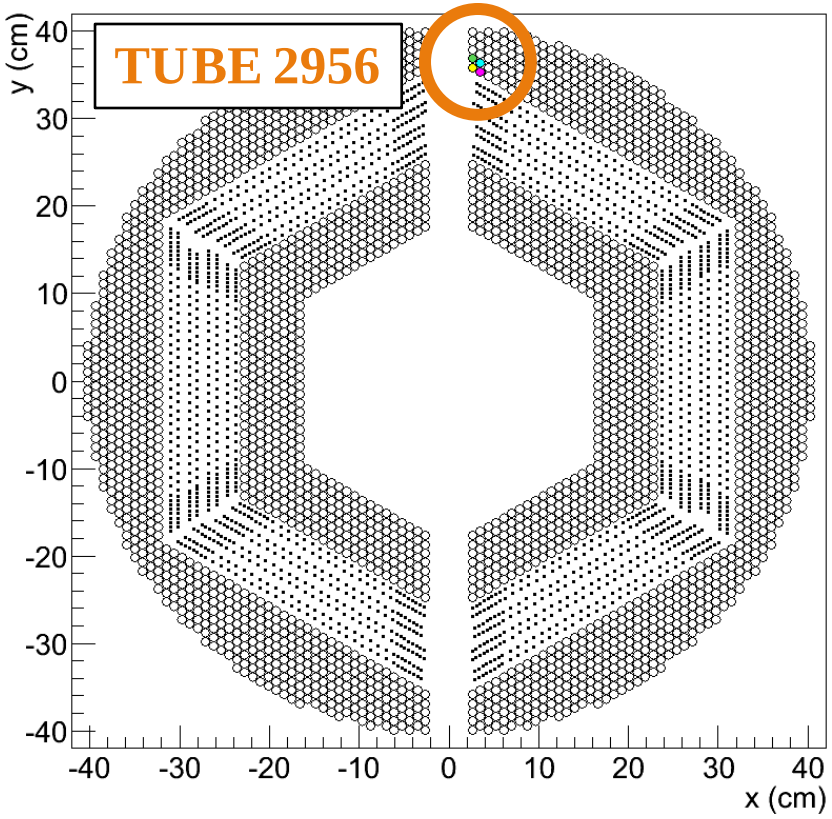
SECTORS



New functions for STT

In this way it is also possible to obtain an array of neighbouring tubes

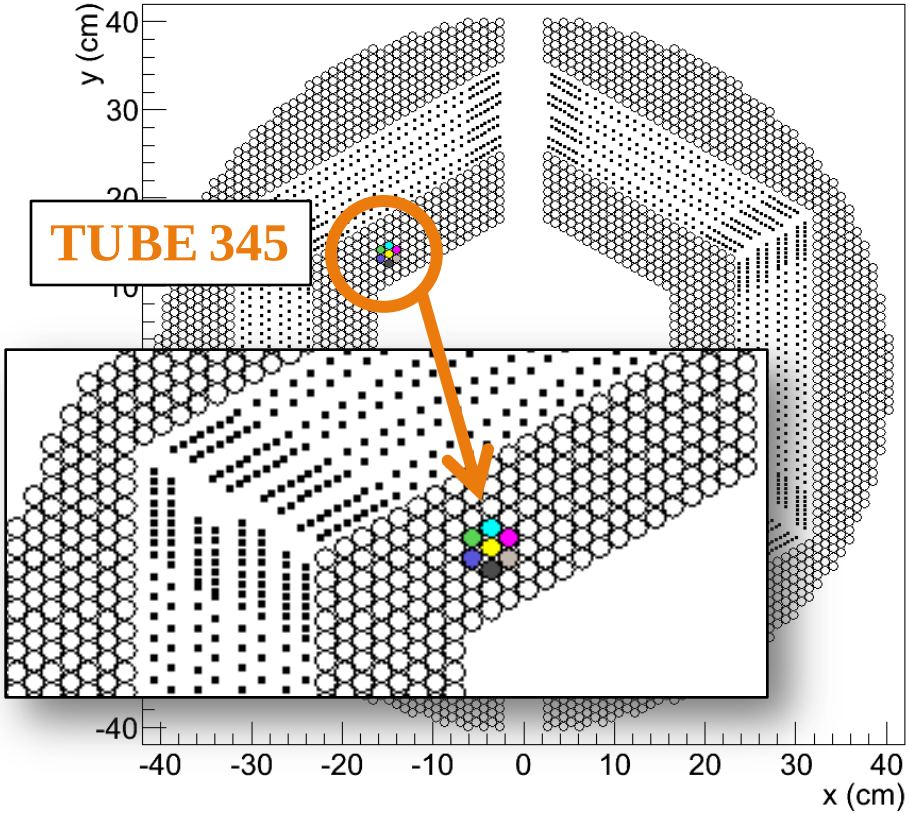


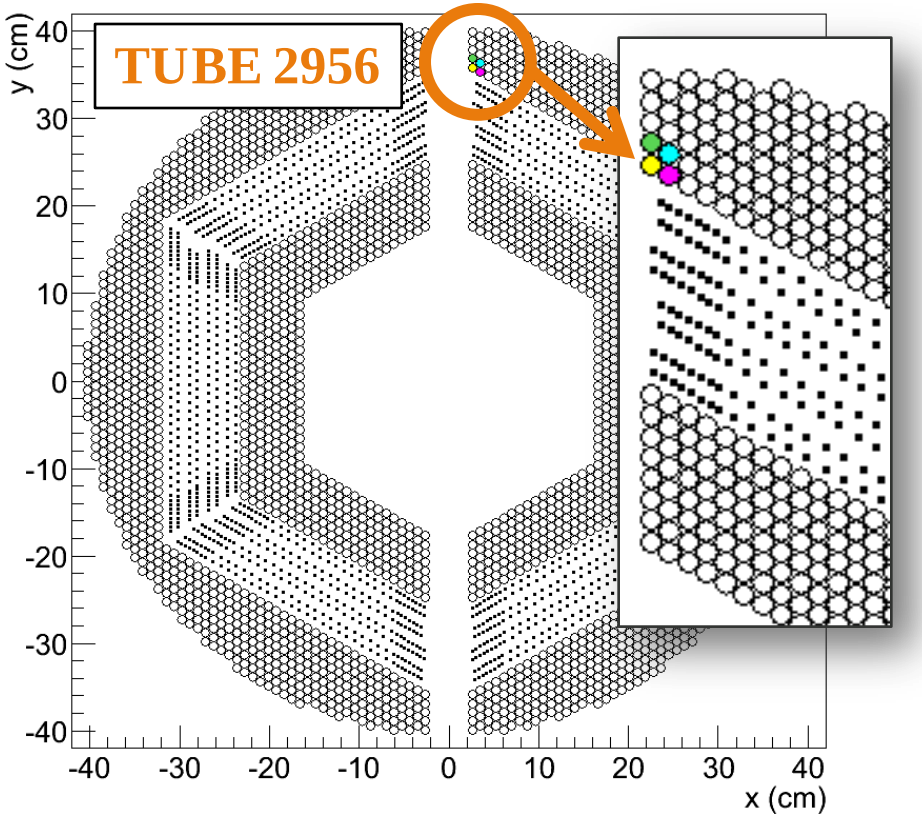


… These functions should help in the cleanup of the found tracks

New functions for STT

In this way it is also possible to obtain an array of neighbouring tubes





… These functions should help in the cleanup of the found tracks

Conclusion

- Structure to integrate primary and secondary track finder
  it should permit the integration of the prim. and sec. track finders
- New implementation for secondary track finder
  it has to be completed for prim. and then applied to the sec. tracks
- New tool for STT
  this should help the cleanup and/or the clustering procedure

Thank you