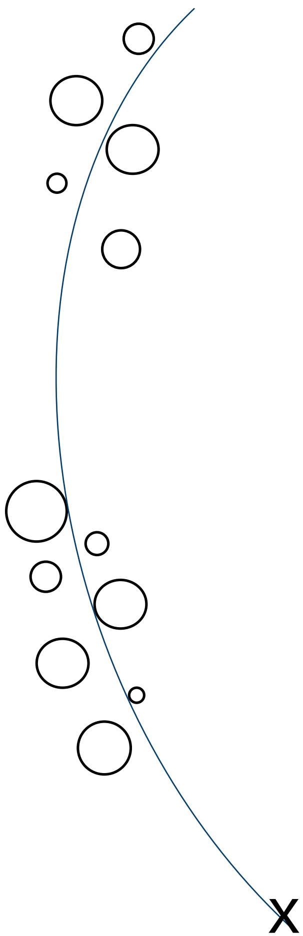
UPDATE TO THE APOLLONIUS HOUGH TRACK FINDER

05.11.2019 I ANNA SCHOLL

JÜLICH
Forschungszentrum

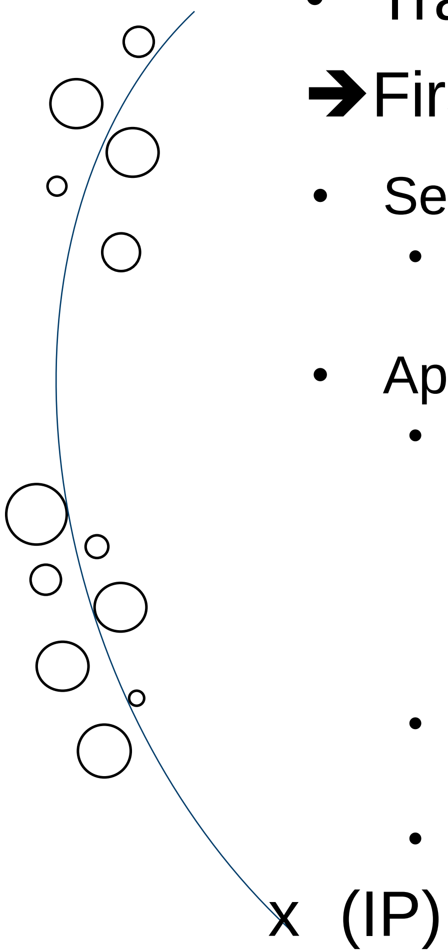
INTRODUCTION



- Implement track finding algorithm for barrel part
- Use hits from MVD, STT, GEM detector
- Track passes through MVD and GEM hit points
- Track is tangent to STT isochrones

x interaction point (IP)

HOW TO INCLUDE ISOCHRONE INFORMATION IN TRACKING ALGORITHMS?



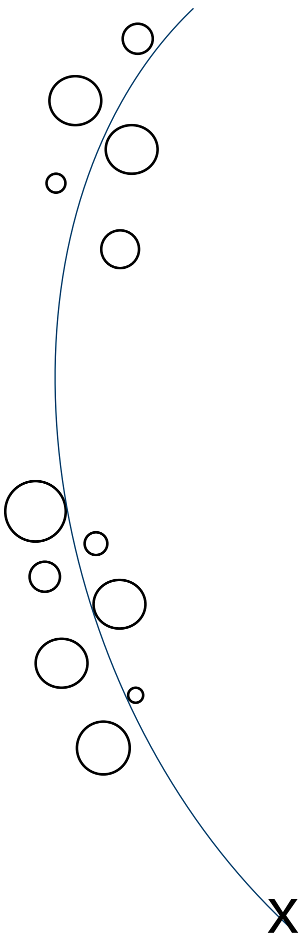
- Track is tangent to the isochrone

➔First idea: **Hough transformation**

- Separate dimensions
  - 3D helix (R, φ, z) ➔ 2D circle (R, φ) + line (z)

- Apply Hough transform to detect tracks in a set of hits
  - For each hit, generate all possible tracks compatible with it (Circles in xy plane, passing through IP and are tangent to the isochrone)
  - Collect generated track parameters for all hits (2D Hough Space)
  - Count: most frequent values = parameters of actual tracks

HOW TO INCLUDE ISOCHRONE INFORMATION IN TRACKING ALGORITHMS?



- Track is tangent to the isochrone

➔First idea: **Hough transformation**

- Separate dimensions
  - 3D helix (R, φ, z) ➔ 2D circle (R, φ) + line (z)

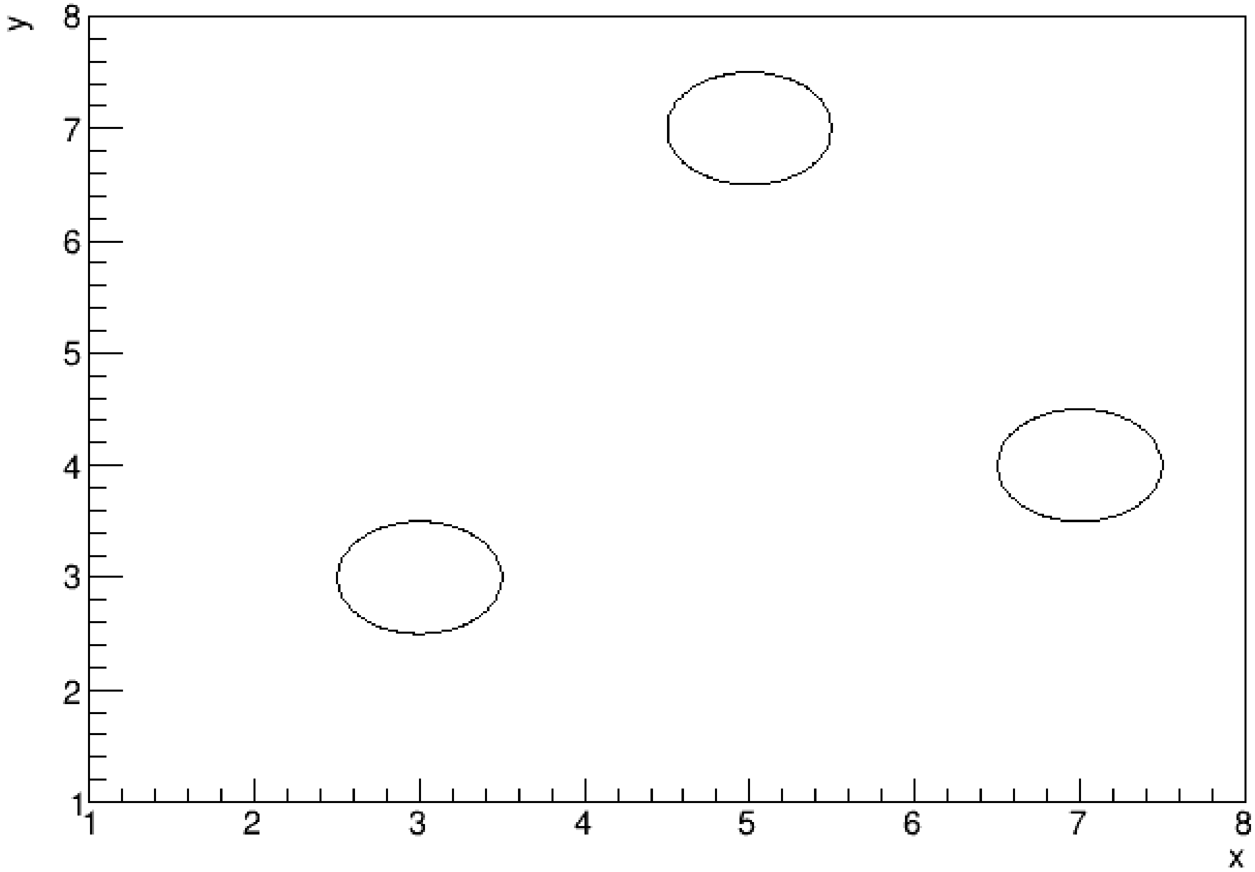
- Apply Hough transform to detect tracks in a set of hits
  - **Problem:** a lot of false combinations for increasing number of tracks per event

  - **Idea:** reduce combinatrics by using 2 Isochrones and IP

➔**problem of Apollonius**

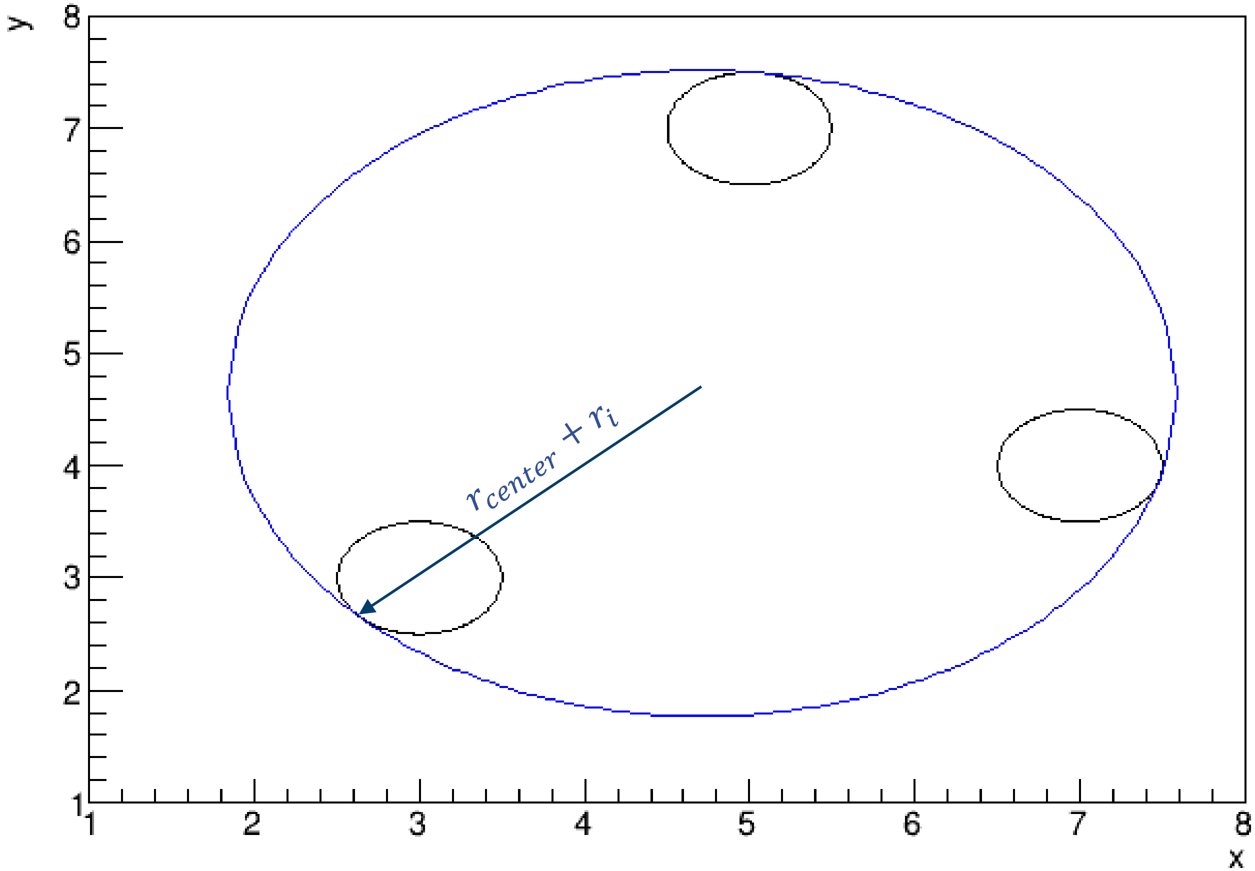
APOLLONIUS PROBLEM

- General Apollonius problem for 3 circles:
  Find circles that are tangent to three given circles in a plane



APOLLONIUS PROBLEM

- General Apollonius problem for 3 circles:
  Find circles that are tangent to three given circles in a plane

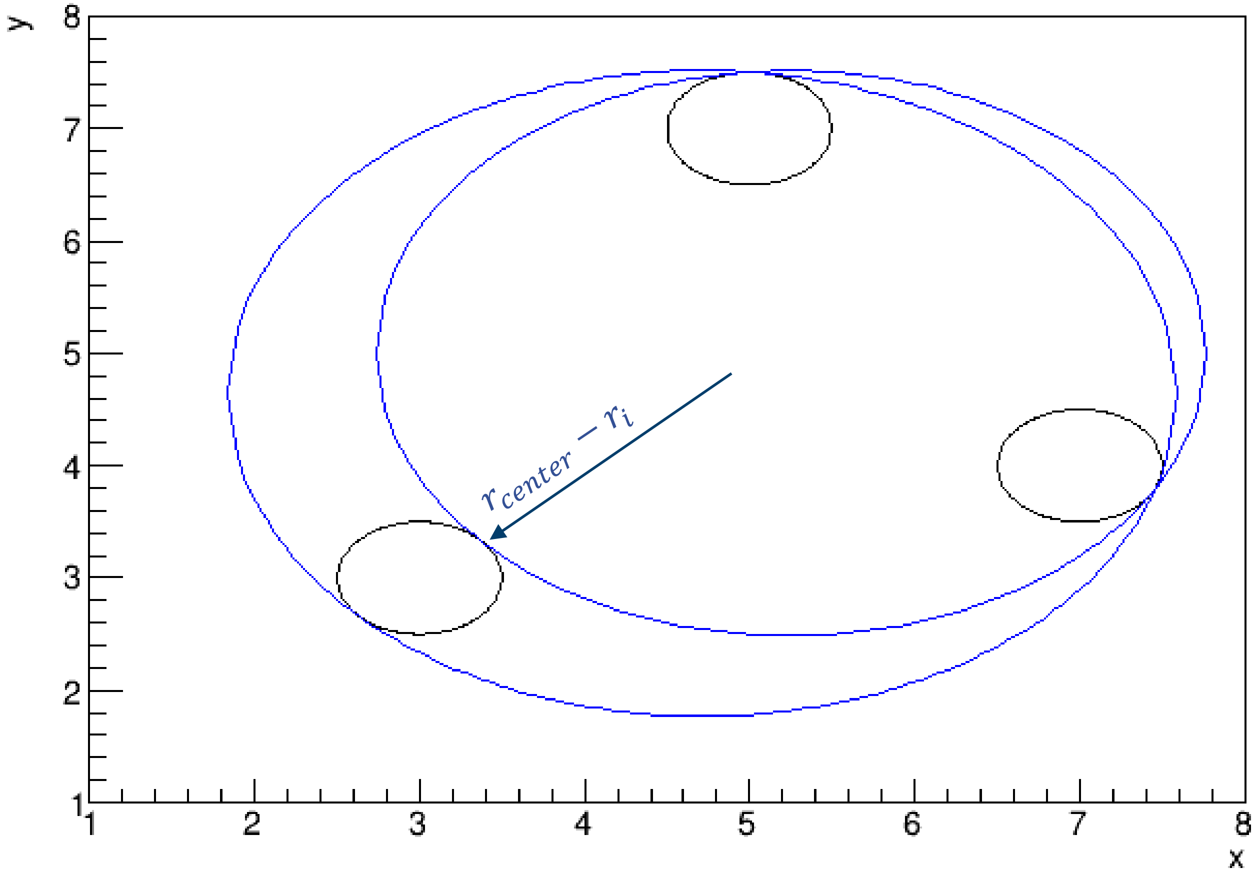


For each circle there are 2 possibilities for an Apollonius circle

1. $r_{Apollonius} = r_{center} + r_i$

APOLLONIUS PROBLEM

- General Apollonius problem for 3 circles:
  Find circles that are tangent to three given circles in a plane

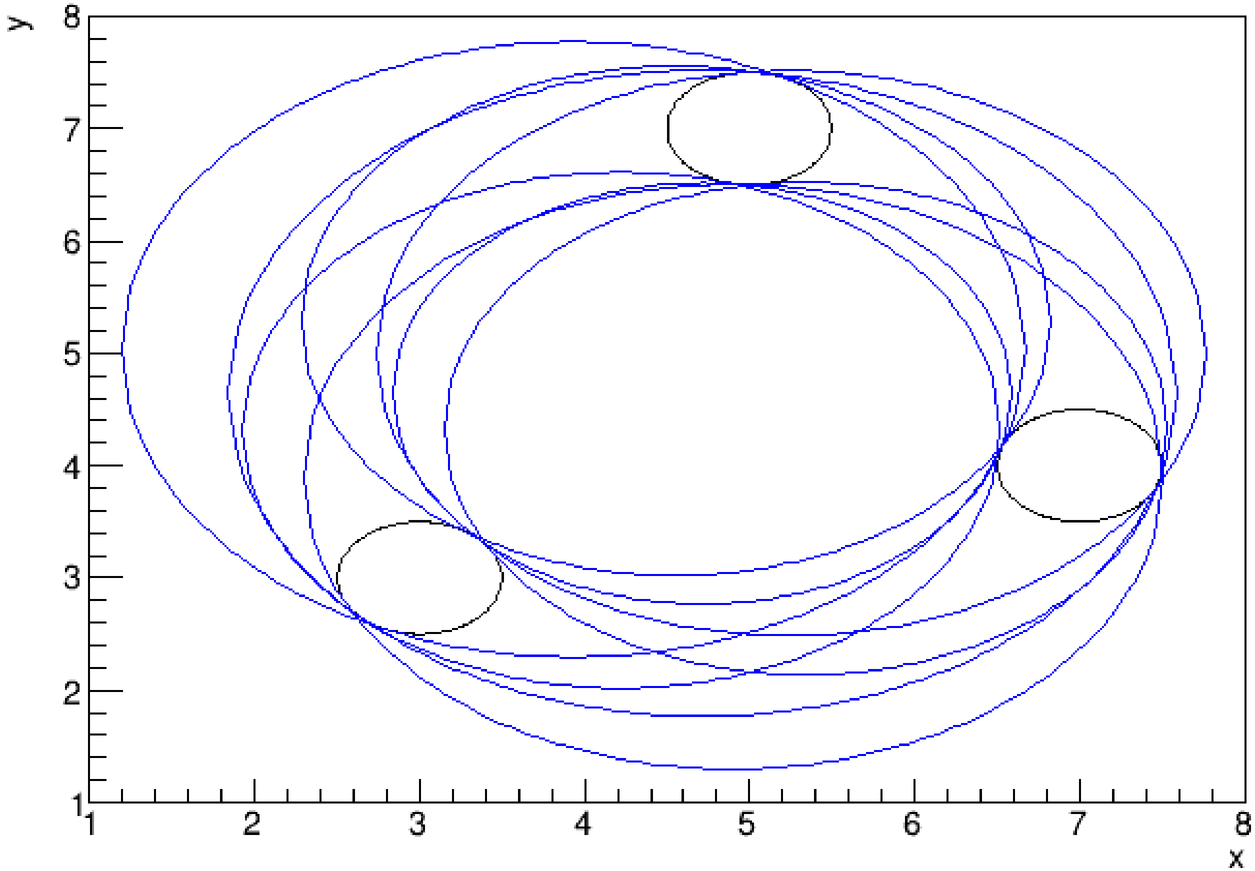


For each circle there are 2 possibilities for an Apollonius circle

1. $r_{Apollonius} = r_{center} + r_i$
2. $r_{Apollonius} = r_{center} - r_i$

APOLLONIUS PROBLEM

- General Apollonius problem for 3 circles:
  Find circles that are tangent to three given circles in a plane



For each circle there are 2 possibilities for an Apollonius circle

1. $r_{Apollonius} = r_{center} + r_i$
2. $r_{Apollonius} = r_{center} - r_i$

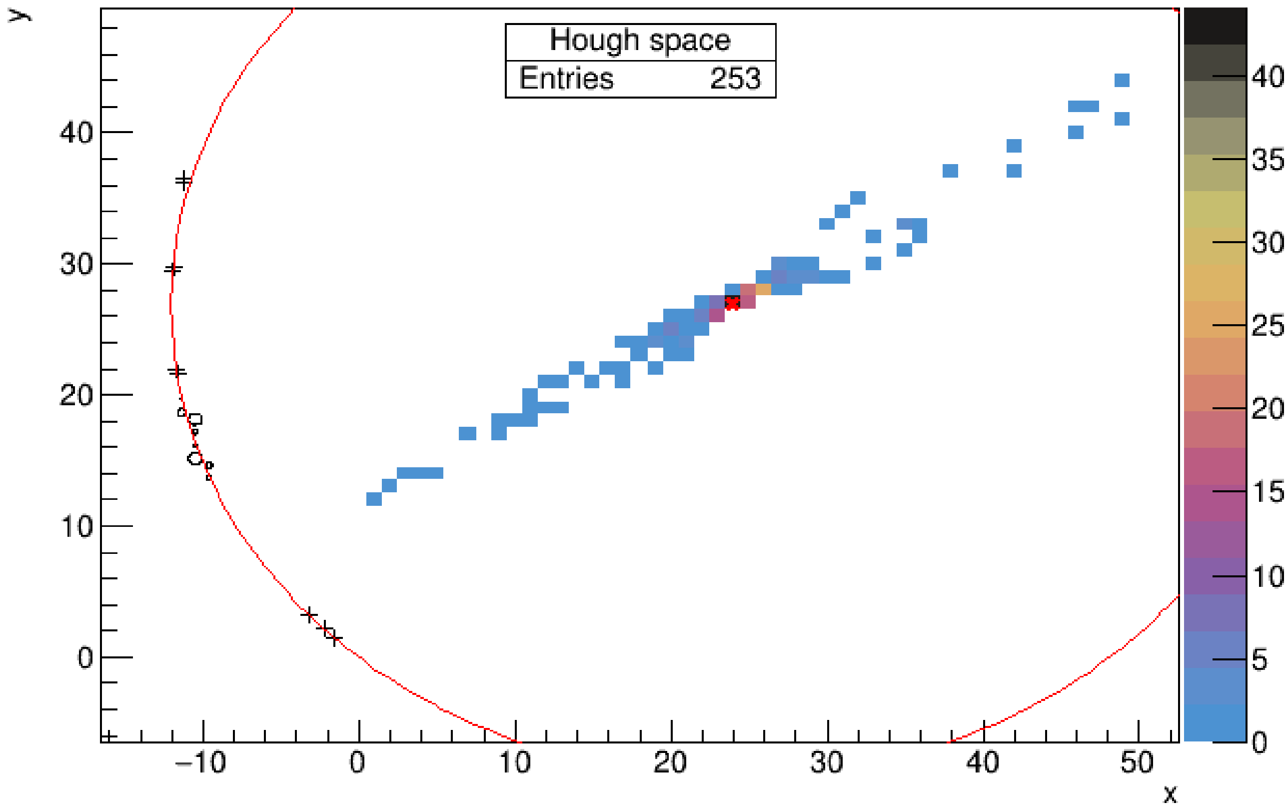
In total $2^3 = 8$ Apollonius circles

JÜLICH Forschungszentrum

HOUGH TRANSFORMATION BASED ON THE APOLLONIUS PROBLEM

Implemetation in PandaRoot and testing with simulated data

Example for one Track

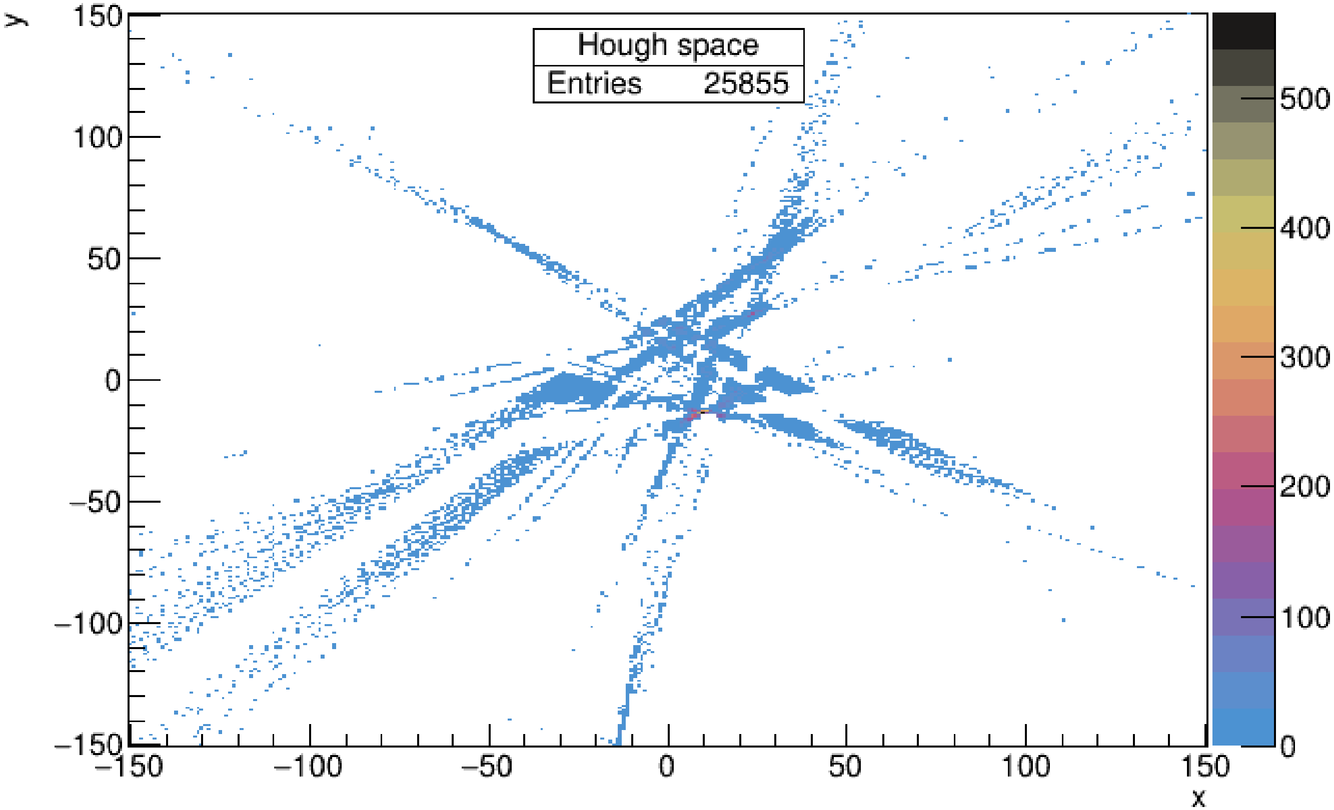


- Works quiet well if track candidate is known (IdealTrackFinder)

HOUGH TRANSFORMATION BASED ON THE APOLLONIUS PROBLEM

Implemetation in PandaRoot and testing with simulated data

Example for one Event



- Works quiet well if track candidate is known (IdealTrackFinder)
- For one event (many tracks): high combinatorics with (still) many false combinations

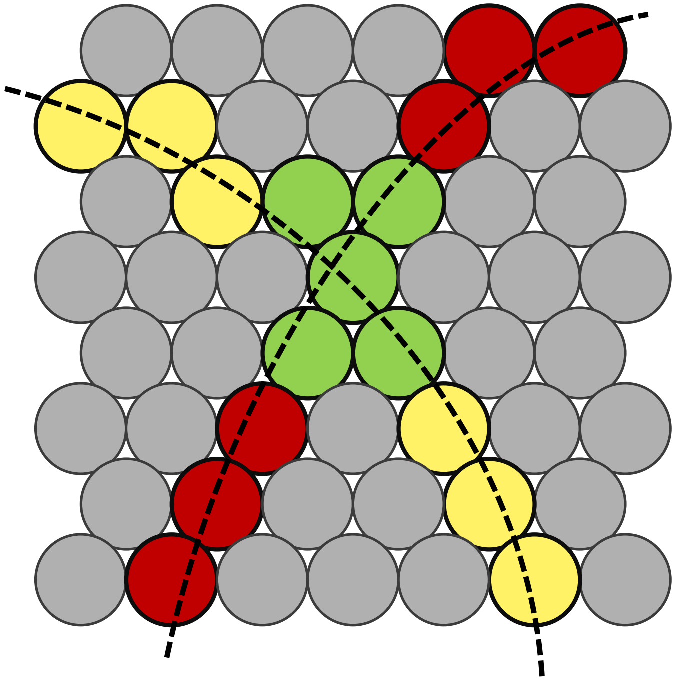
PRESELECTION

- Using Apollonius transform for all tracks in one event is very time consuming and leads to a lot of false combinations

➔ preselection for possible tracklets is needed

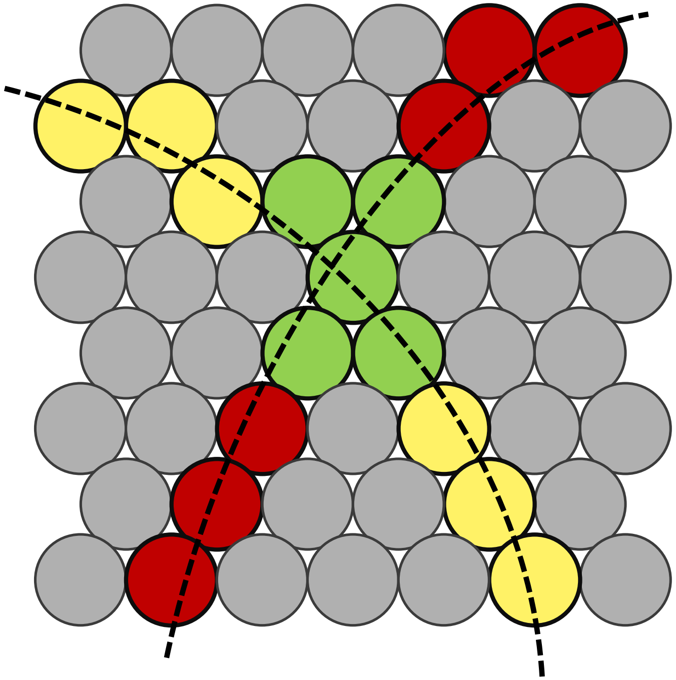
- Idea:
  - preselection by cellular automaton

CELLULAR AUTOMATON



- Combine directly adjacent hits to tracklets

CELLULAR AUTOMATON



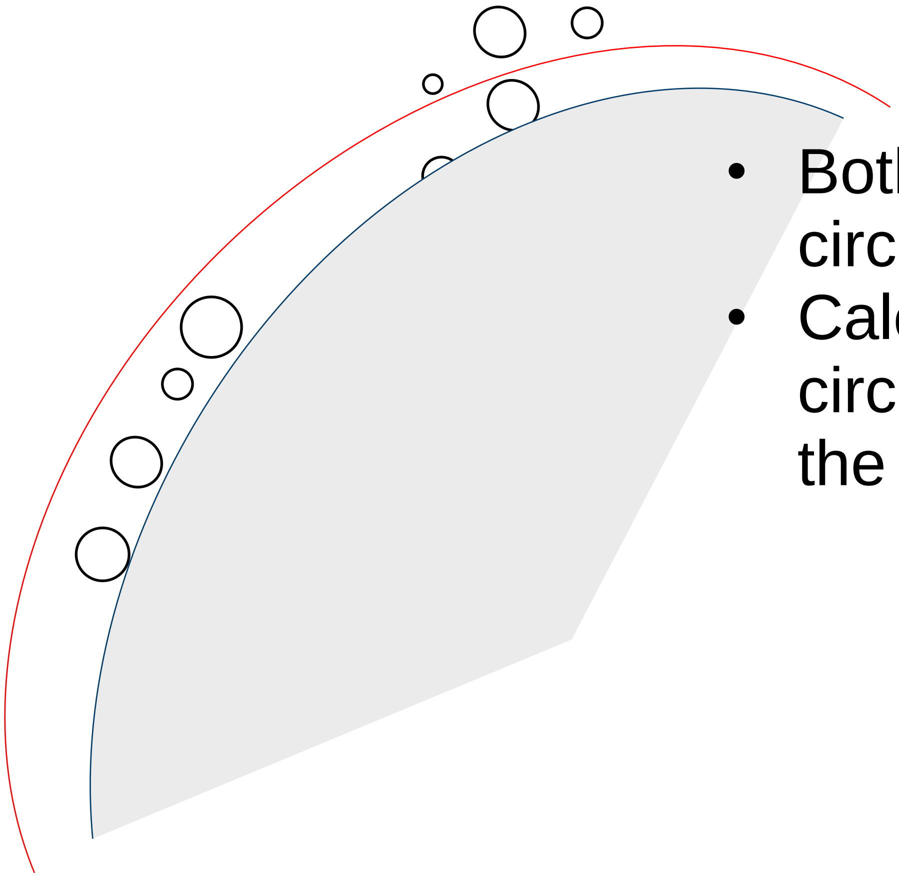
- Combine directly adjacent hits to tracklets
- Problem: Tracks can be divided in several tracklets

➔ How to merge tracklets
  ➔ Find relevant parameters for merging

MERGING TRACKLETS

Compare two different parameters for merging:

1. intersection area fraction

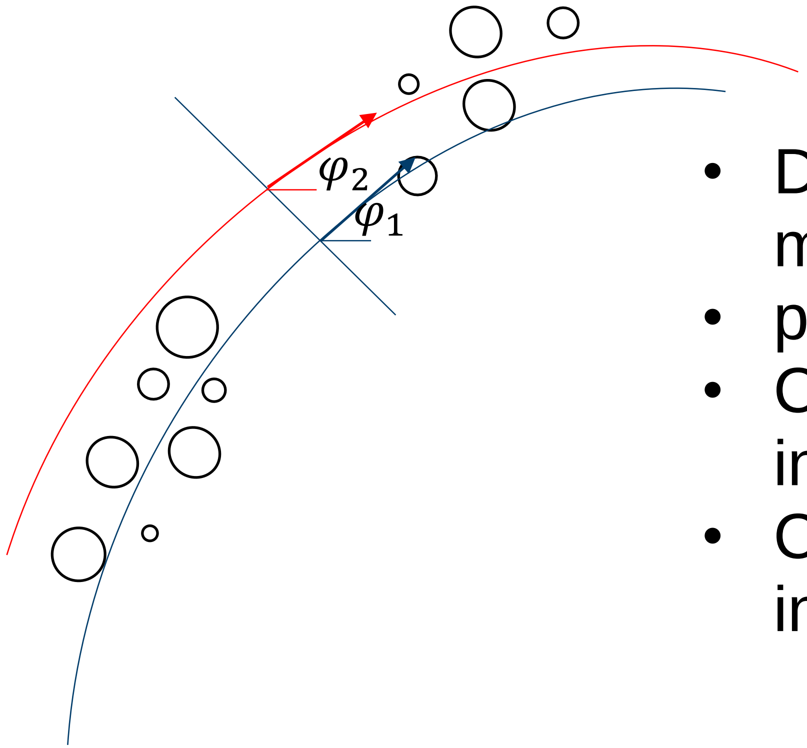


- Both tracks are represented by circles
- Calculate intersection area of both circles and divide it by the area of the larger circle

MERGING TRACKLETS

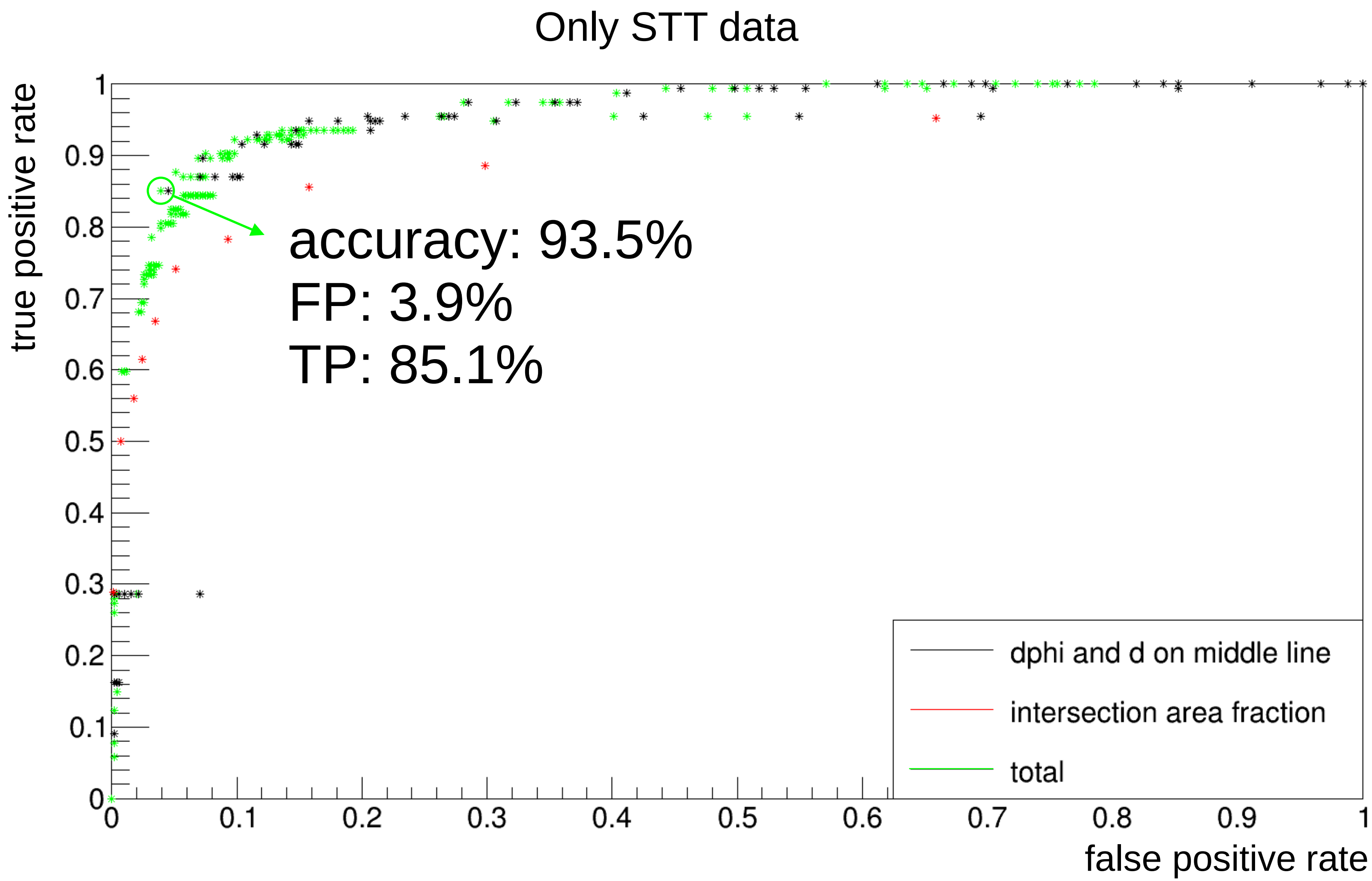
Compare two different parameters for merging:

1. intersection area fraction
2. middle line



- Define line perpendicular in the middle between two tracklets
- propagate both tracks to this line
- Calculate distance between both intersection points
- Calculate angular difference at the intersection points

MERGING TRACKLETS



TESTING ALGORITHM WITH MERGING

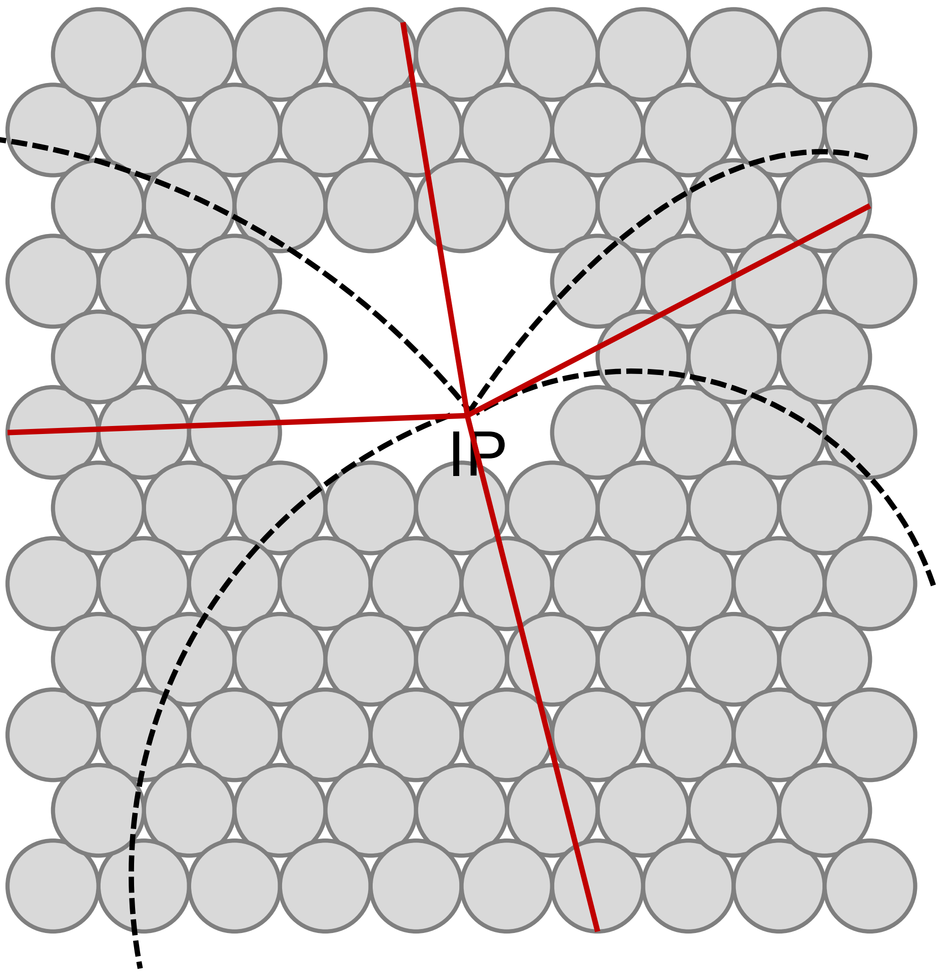
- Applying merging to the algorithm leads still to a high rate of not found tracks (42% not found)
- Reason:
  - after Cellular Automaton and merging tracklets still hits are not added to a track
  - A track is divided in so many small tracklets that the track could not be found

- Searching for tracks in remaining hits
  - leads to high combinatorics and many false combinations
  - Again use a preselection

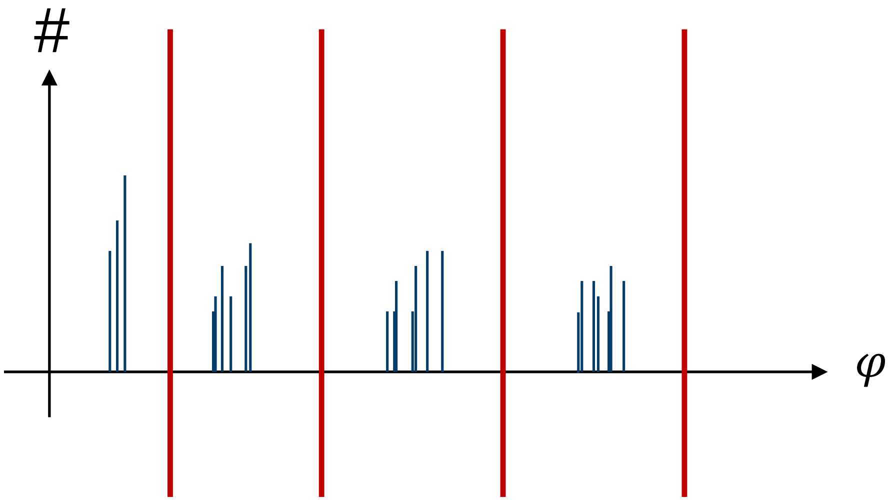
  → dividing remaining hits in Segments and perform hough transformation to hits found with segmentation algorithm

JÜLICH Forschungszentrum

SEGMENTATION ALGORITHM

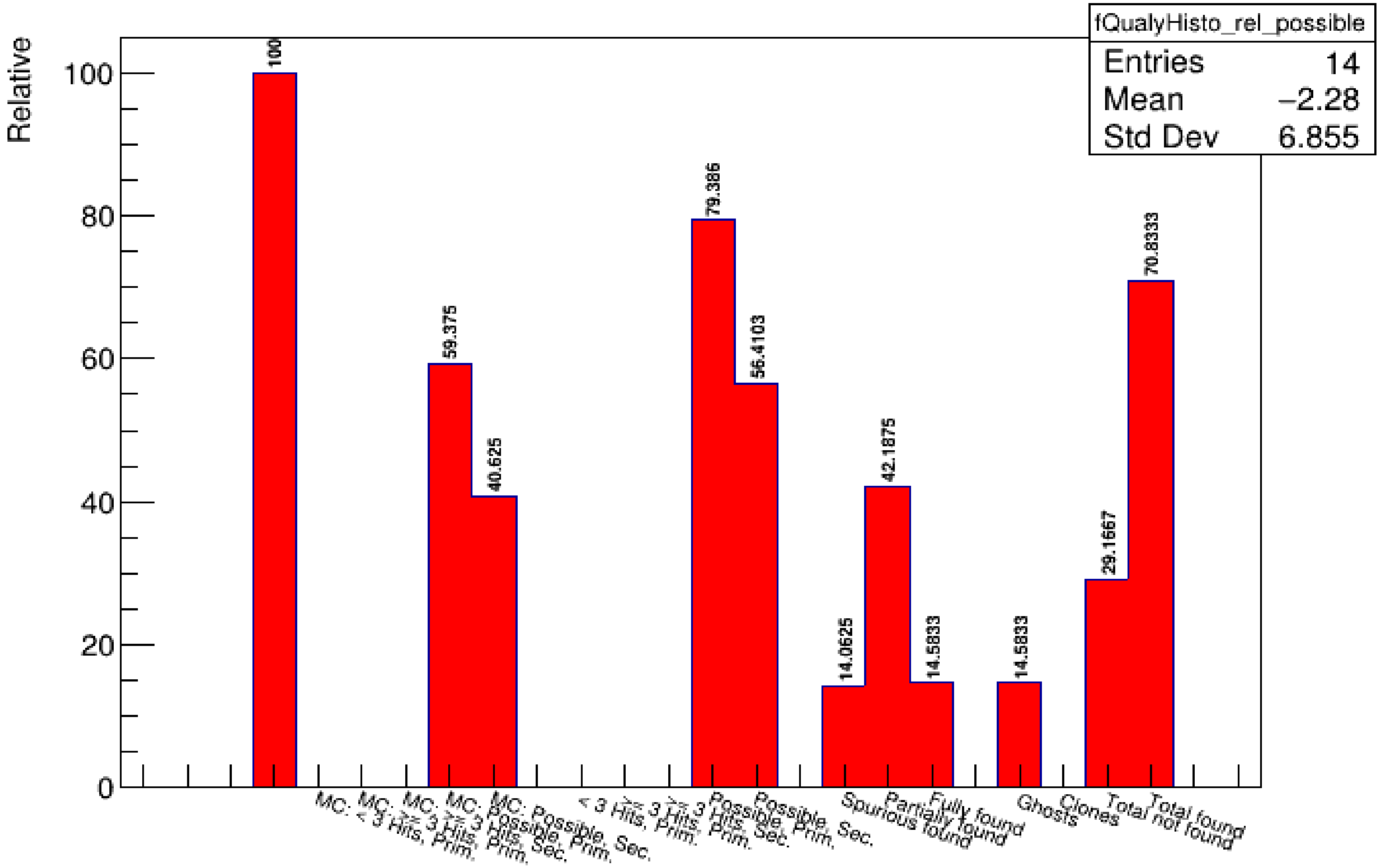


- Filling φ-values of all hits into a histogram:



- Divide in φ - sectors
- Hough transformation for all hits in one sector

RESULTS



JÜLICH Forschungszentrum

NEXT STEPS

- Speed up computing time (at the moment ~0.5 s /event) and enable GPU calculation
- Try to decrease ghost ratio and increase number of fully found tracks
- Insert z-direction