JÜLICH
FORSCHUNGSZENTRUM

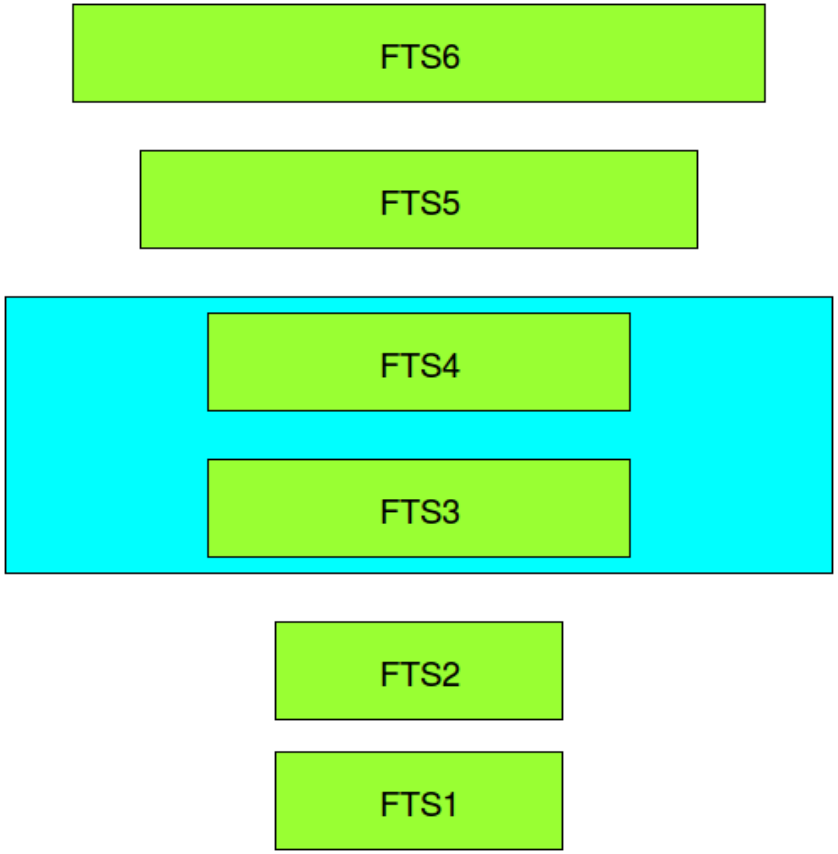
Implementation of a trackfinding-algorithm for the forward tracking system of the PANDA-Detector

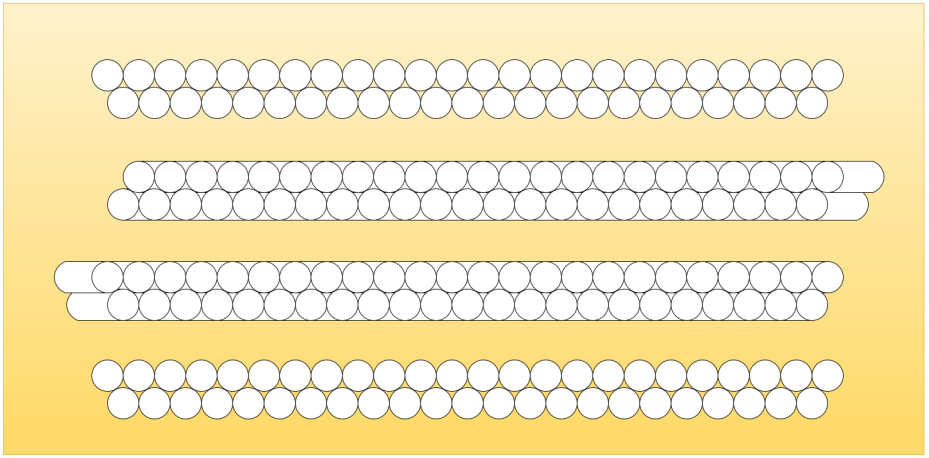
13th September 2016 | Felix Kibellus - IKP

Overview

- Introduction to the FTS
- Presentation of the algorithm
  - Find track-candidates
  - Approximate lines
  - Combine lines inside one FTS-station
  - Combine lines between FTS-stations
  - Trackfinding inside the magnetic field
  - Adding unassigned Hits
- Quality analysis
- Summary and outlook

Introduction to the FTS



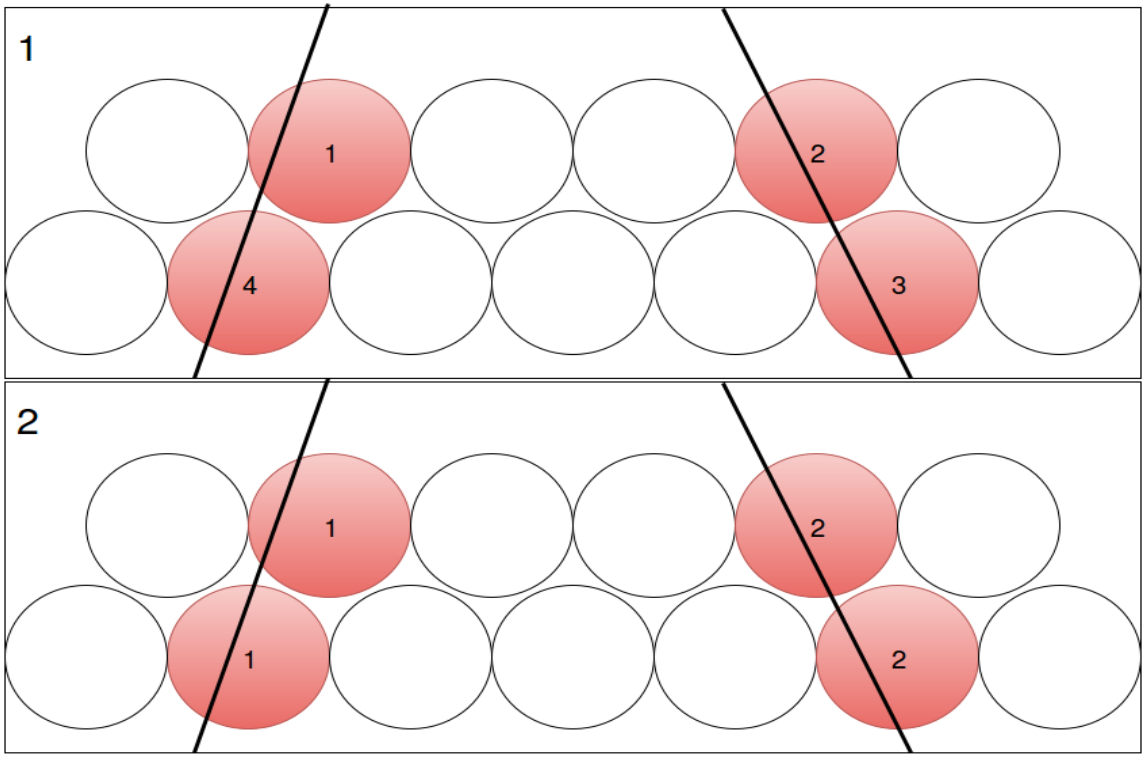


- 6 FTS-stations
- Magnetic field between FTS2 and FTS5

- Each station consists of 4 double-layer straw-tubes
- Second layer is skewed 5° to right
- Third layer is skewed 5° to left
- First and last layer are not skewed

Creating track-candidates

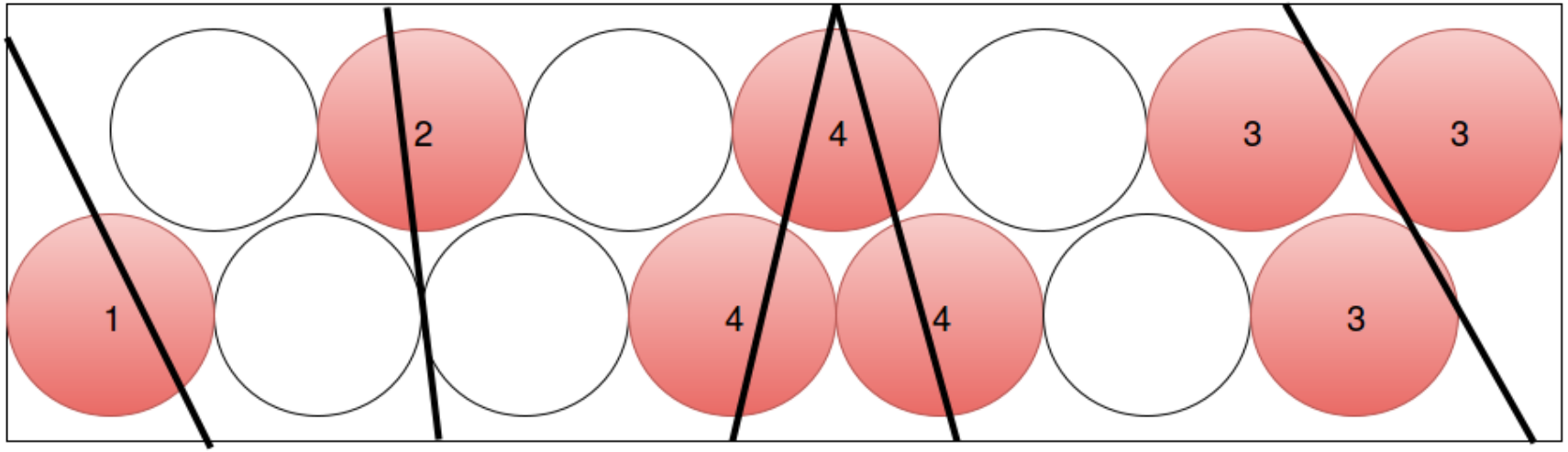
Use a cellular automaton to group FTS-hits



1. Initialize the state of each hit with the unique tube-ID
2. Set the state to the minimum of the own state and the state of the neighbors
3. Refresh step 2 while states are changing

4. Hits with the same state are part of the same track

Creating track-candidates

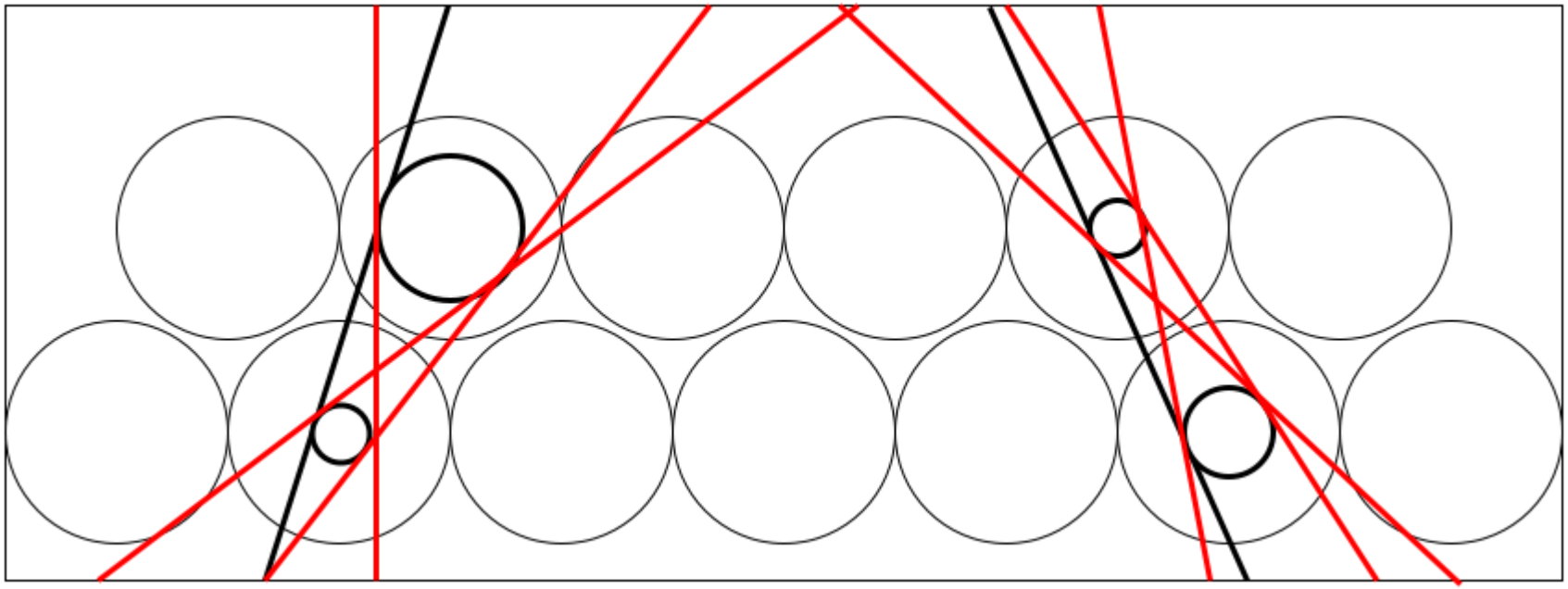


There are different cases:

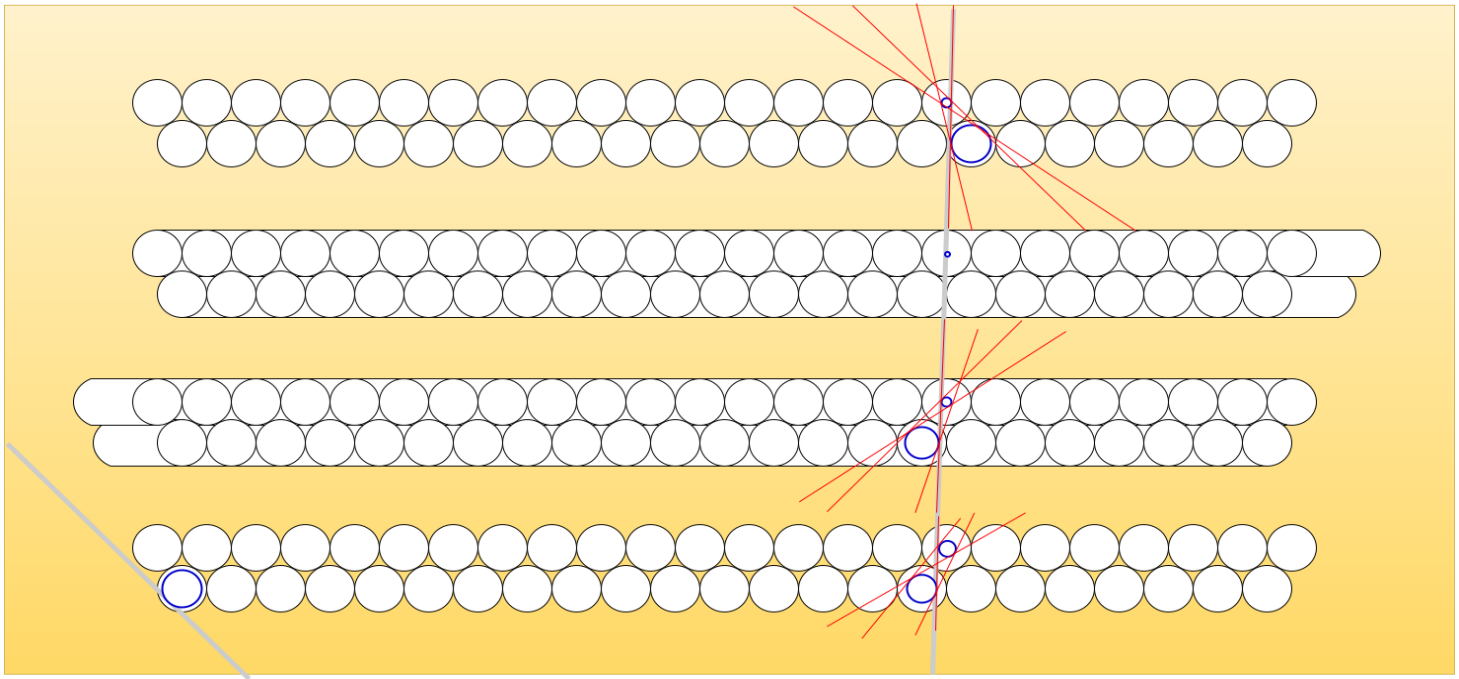
1. Track-candidate consists of only one hit because the track hits only one straw-tube at the edge
2. Only one hit because the track hits no straw-tube in the first layer
3. More than two hits because the angle is pointed
4. More than two hits because two tracks are crossing in the double layer

Approximate lines

Approximate the 4 lines for each tracklet with the isochrones



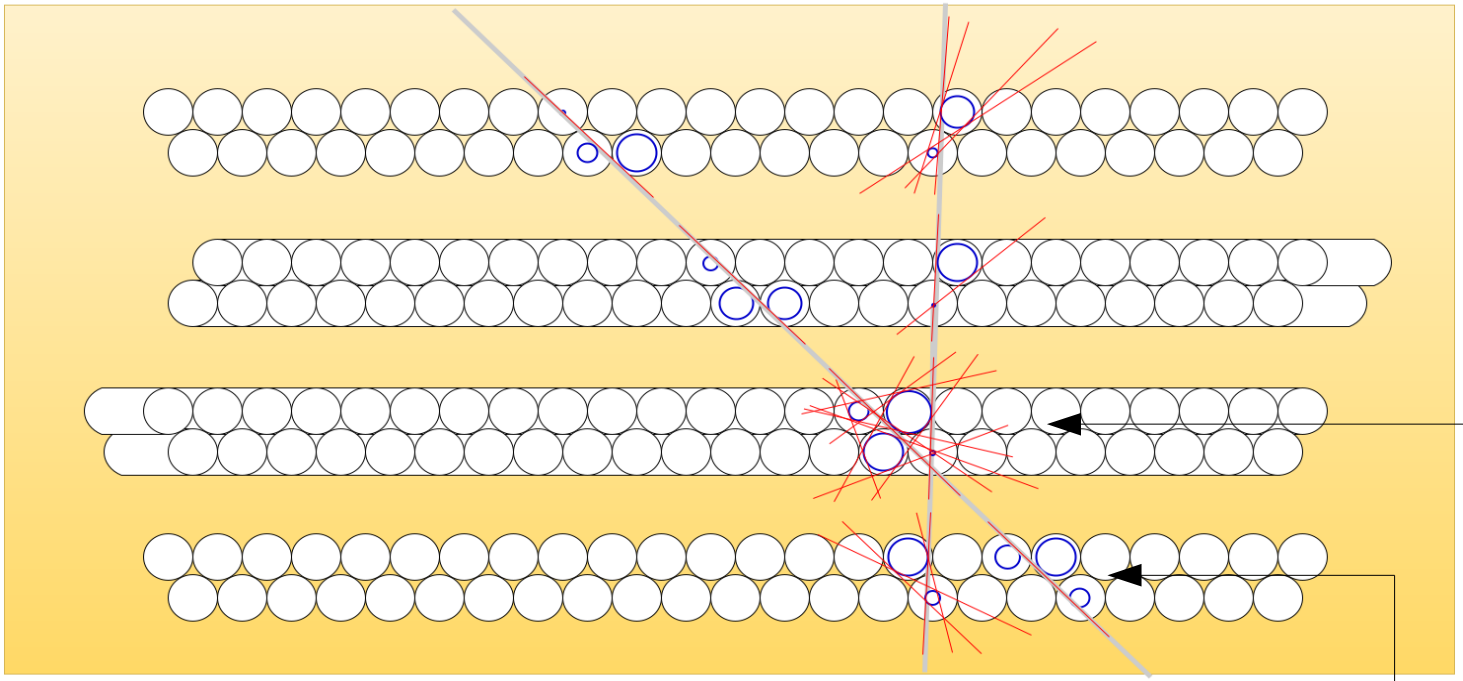
Approximate lines – special cases



Only one hit in track-candidate: no lines can be approximated
Two hits in track-candidate: 4 lines can be approximated

Two hits is the standard case because the likelihood for crossing exactly two straw-tubes is the highest

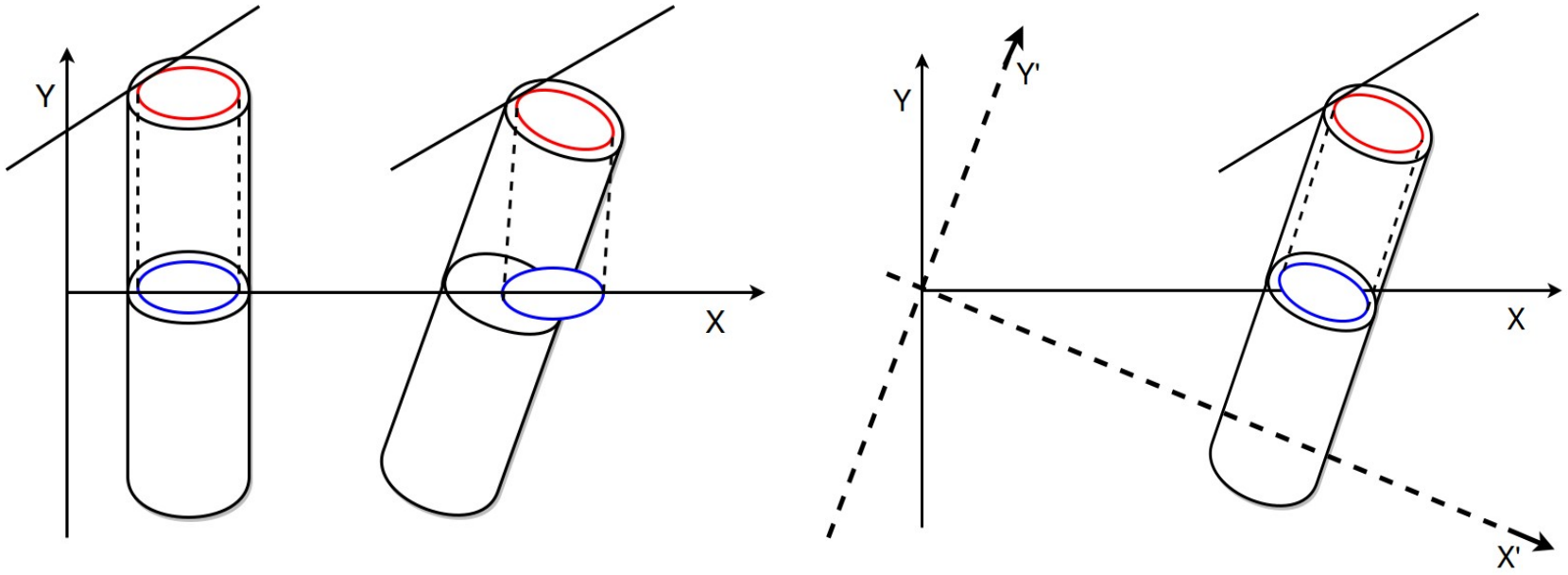
Approximate lines – special cases



More than two hits in track-candidate:
Two Cases

- **Case 1**
  unique line can be found: choose the unique line
- **Case 2**
  there is no unique line: create all possible lines

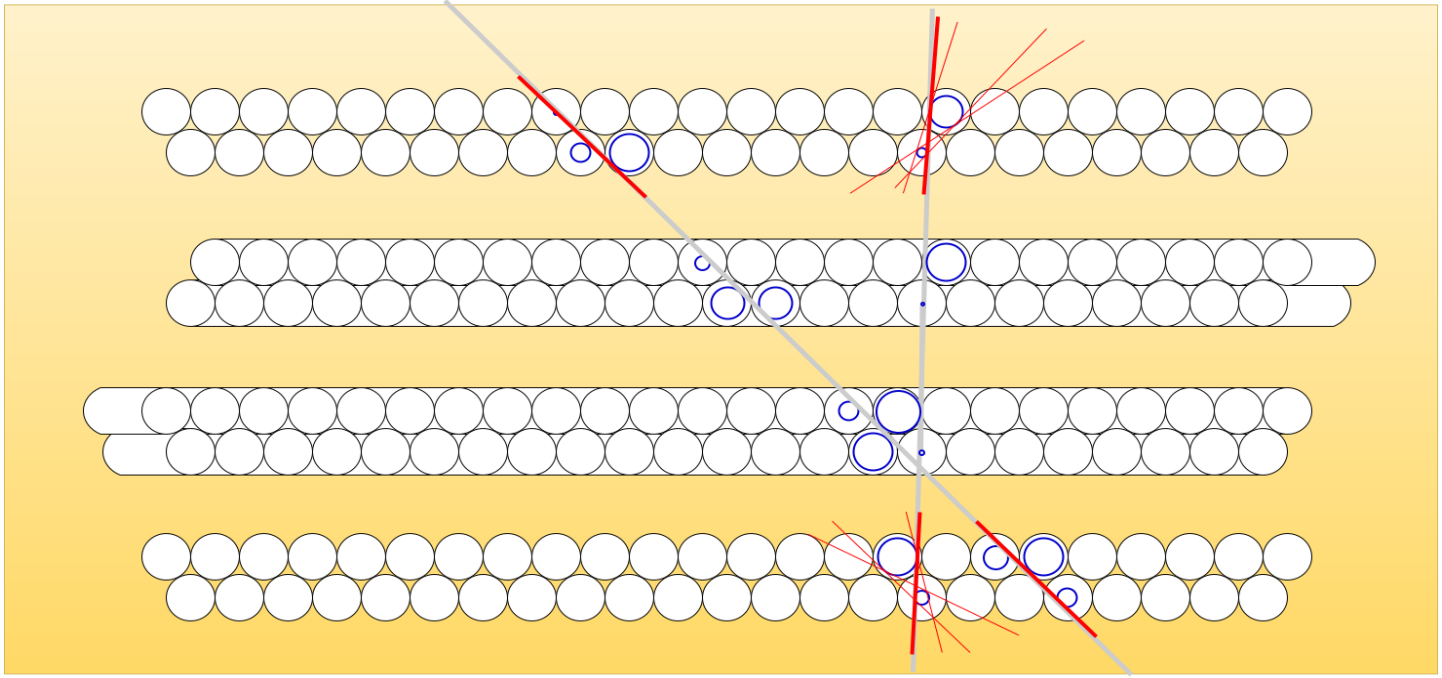
Approximate lines



Problem with skewed straw-tubes:
The projection of the isochrone is depending on the Y-height of the track

=> Coordinate transformation: Rotate the coordinate system by the angle of the straw-tube

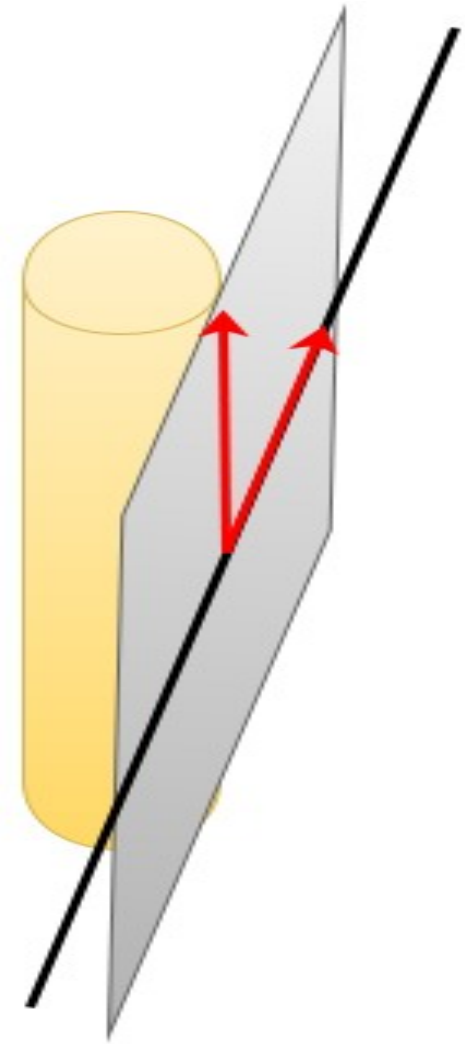
Combine lines inside one FTS-station



Combine layer 1 and 4:

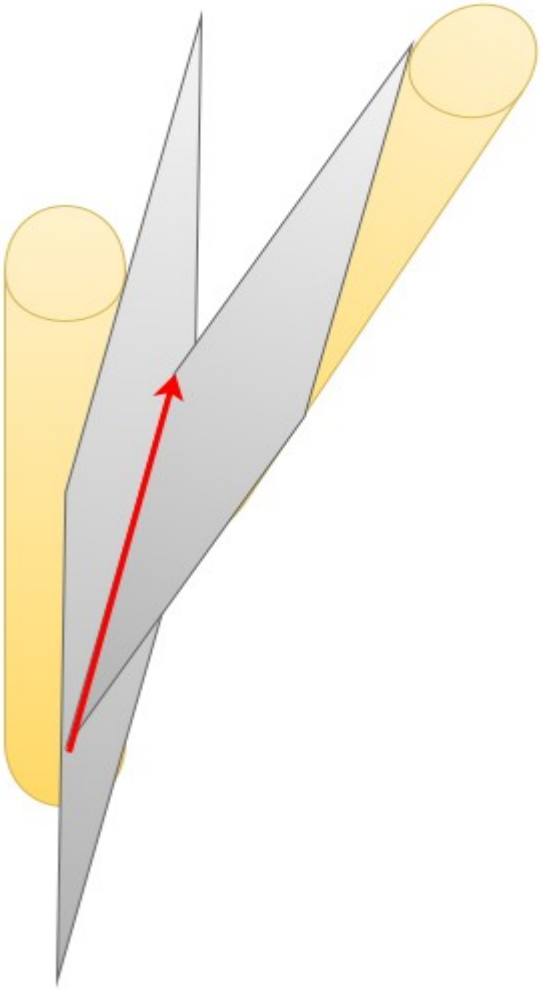
- Search for similar lines
- Similar:
  - Angle between the lines is about 180°
  - Piercing points of the tracks through the middle plane are close to each other
- Combinate layer 1 and 4 to one track

Combine lines inside one FTS-station

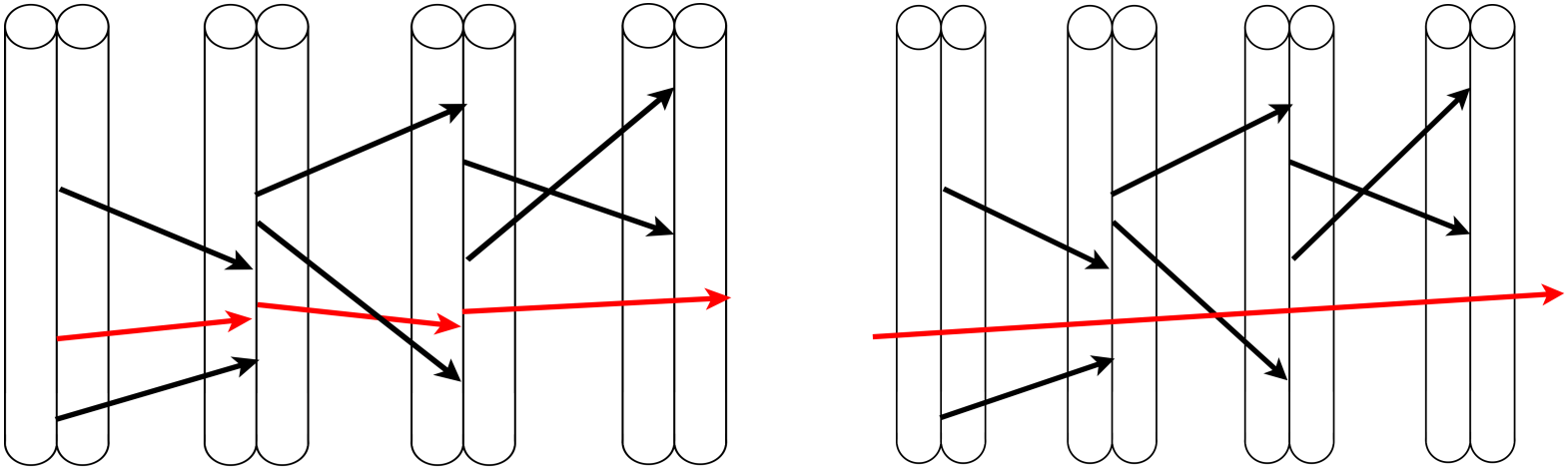


Create planes for each approximation

- First direction vector: approximated line of the step before
- Second direction vector: parallel to the Straw-Tube
- Create intersections of the planes from different FTS-layers



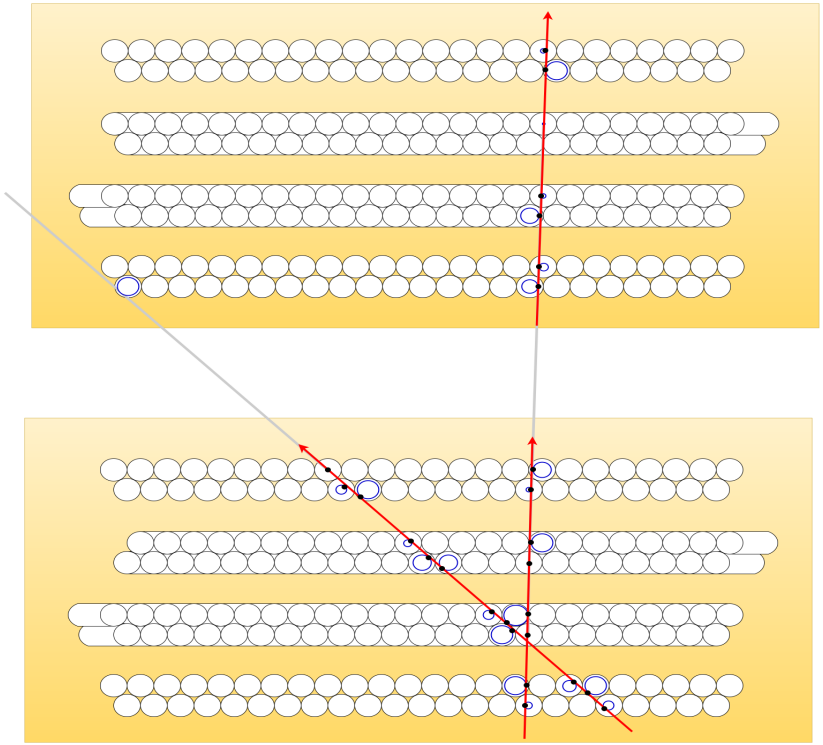
Combine lines inside one FTS-station



The algorithm creates intersection lines

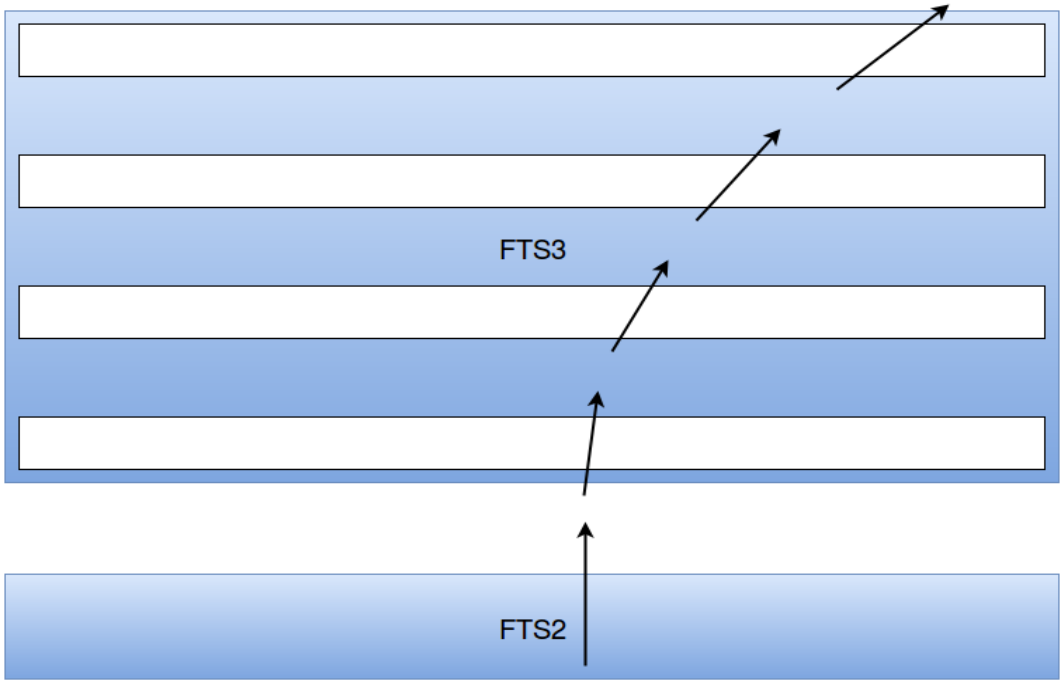
Choose similar lines and create a new approximation with linear regression

Combine lines between FTS-stations



Search again for similar tracks
Create a new approximation which includes both FTS-stations
This algorithm combines FTS1 with FTS2 and FTS5 with FTS6

Trackfinding inside the magnetic field

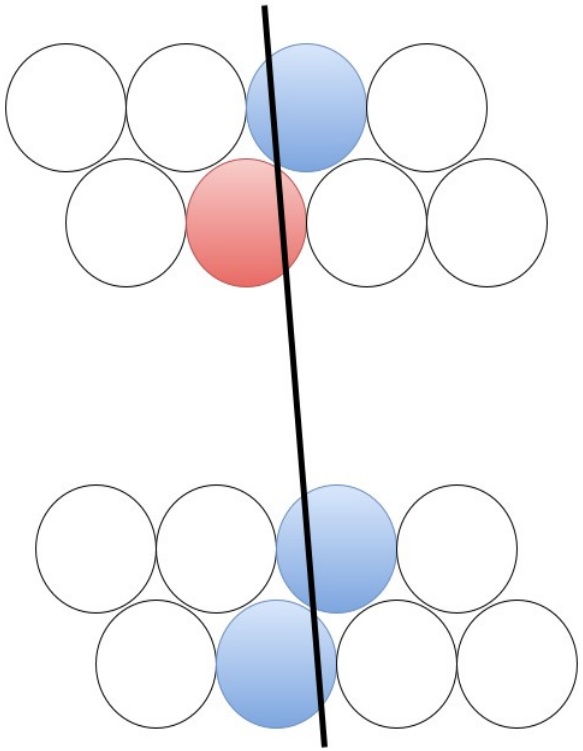


Using a way-follower
for the magnetic field

The algorithm searches the best fitting approximation in the stations 3 and 4

The approximation of FTS1+FTS2 is followed through the magnetic field and will be combined with the approximation of FTS5+FTS6

Adding unassigned hits



After the last step the algorithm reconstructed a complete track

Some hits were not added to the track

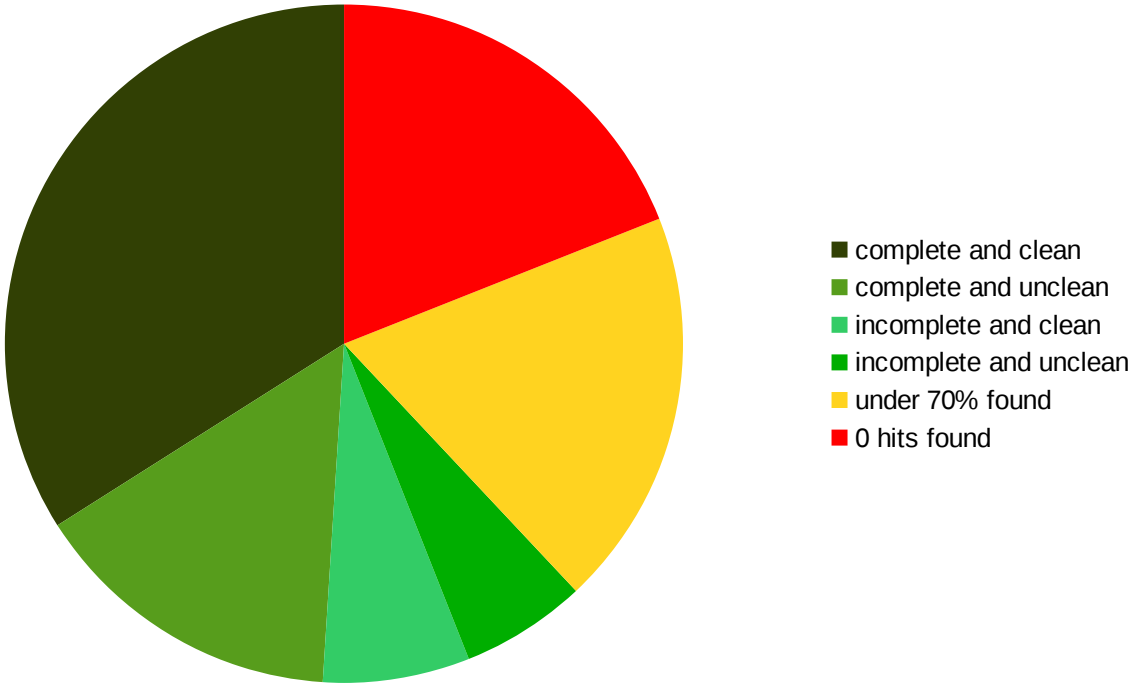
If the distance between the hit and the track is small the hit will be added afterwards

Quality analysis

Used categories:

- Track found (>70% of the hits)
  - Complete and clean:
    contains all hits of the right track, contains no hits from other tracks
  - Complete and unclean:
    contains all hits of the right track, contains hits from other tracks
  - Incomplete and clean
    some hits are missing, contains no hits from other tracks
  - Incomplete and unclean
    - some hits are missing, contains hits from other tracks
- Track not found
  - Less than 70% of the hits
  - 0 hits found

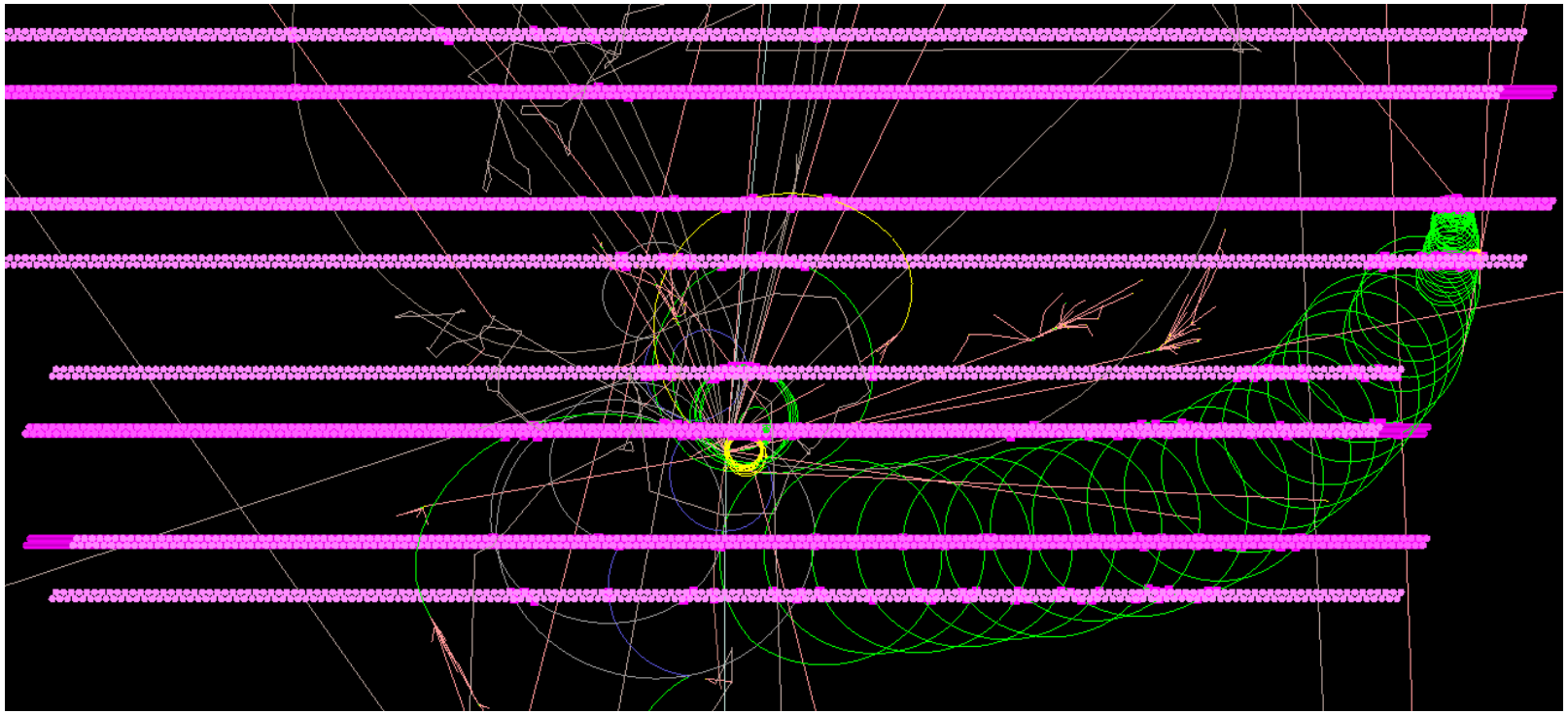
Quality analysis



62% of all tracks could be reconstructed
34% of all tracks were complete and clean

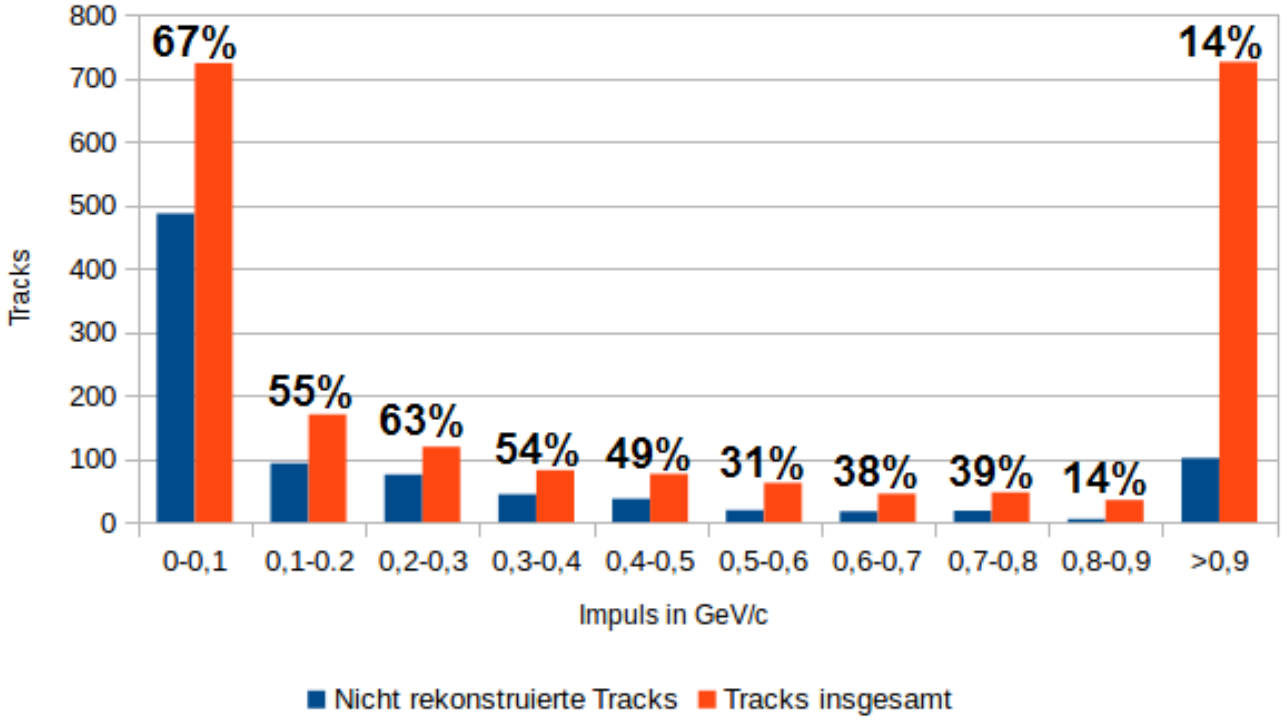
Why are there 38% not reconstructed tracks?

Quality analysis – tracks with low momentum



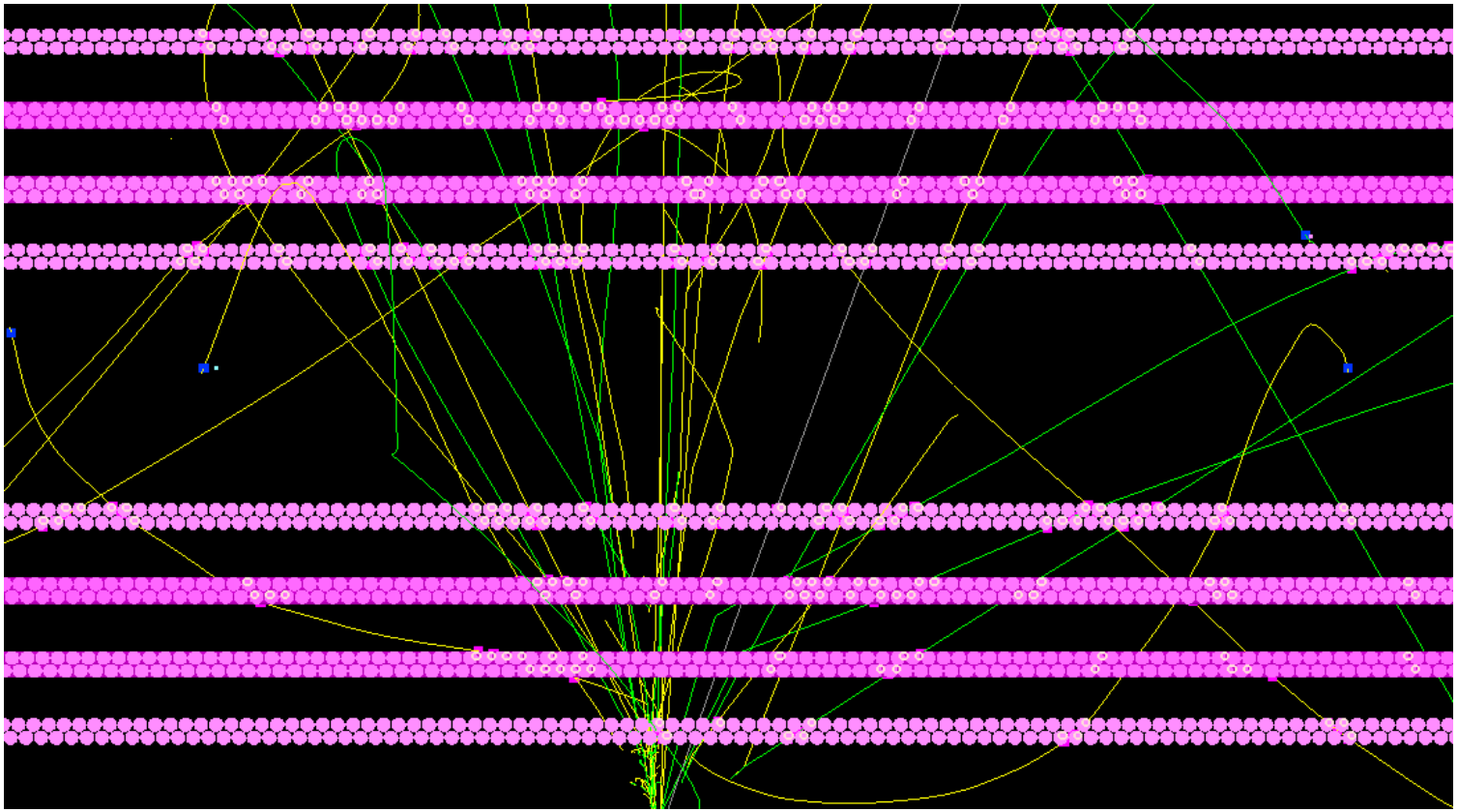
The algorithm has some problems with rotating particles because it can only find tracks heading straight forward

Quality analysis – tracks with low momentum



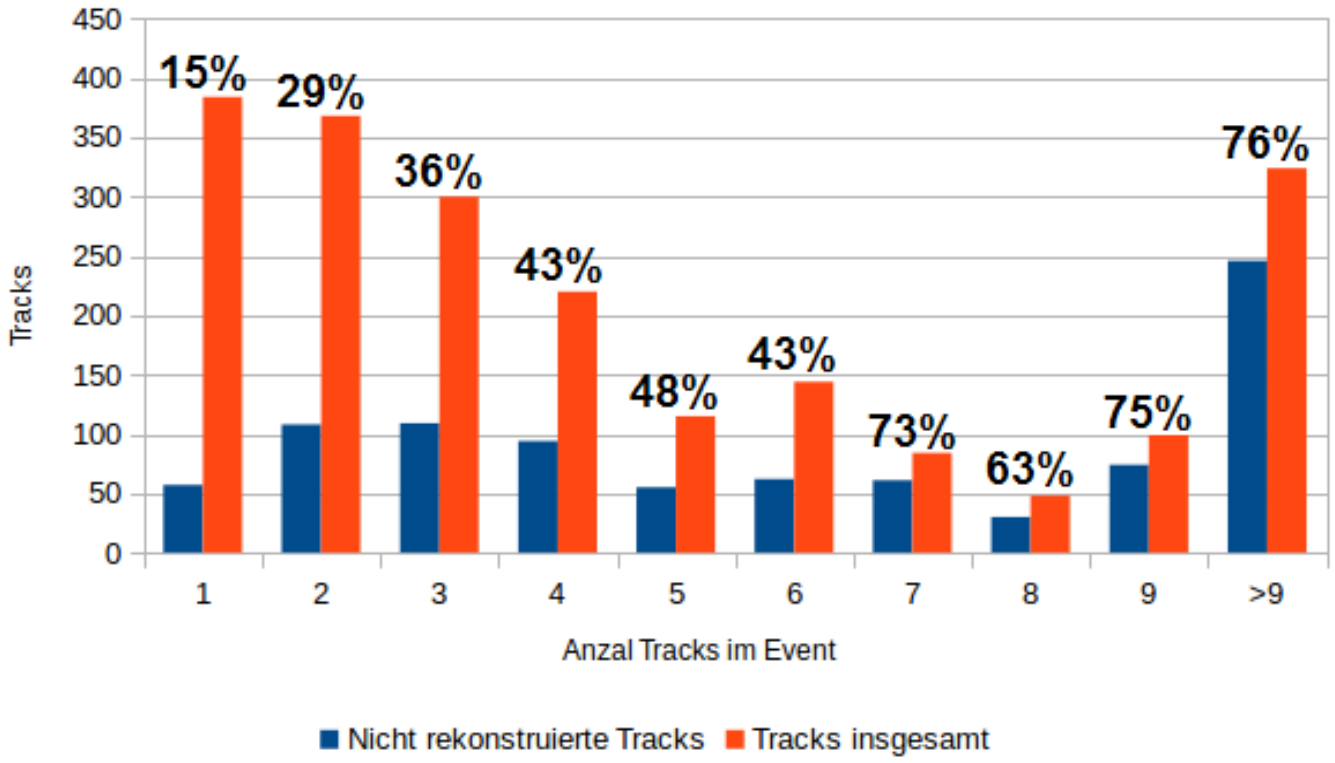
=> There is a correlation between low momentum and not reconstructed tracks

Quality analysis – events with many tracks



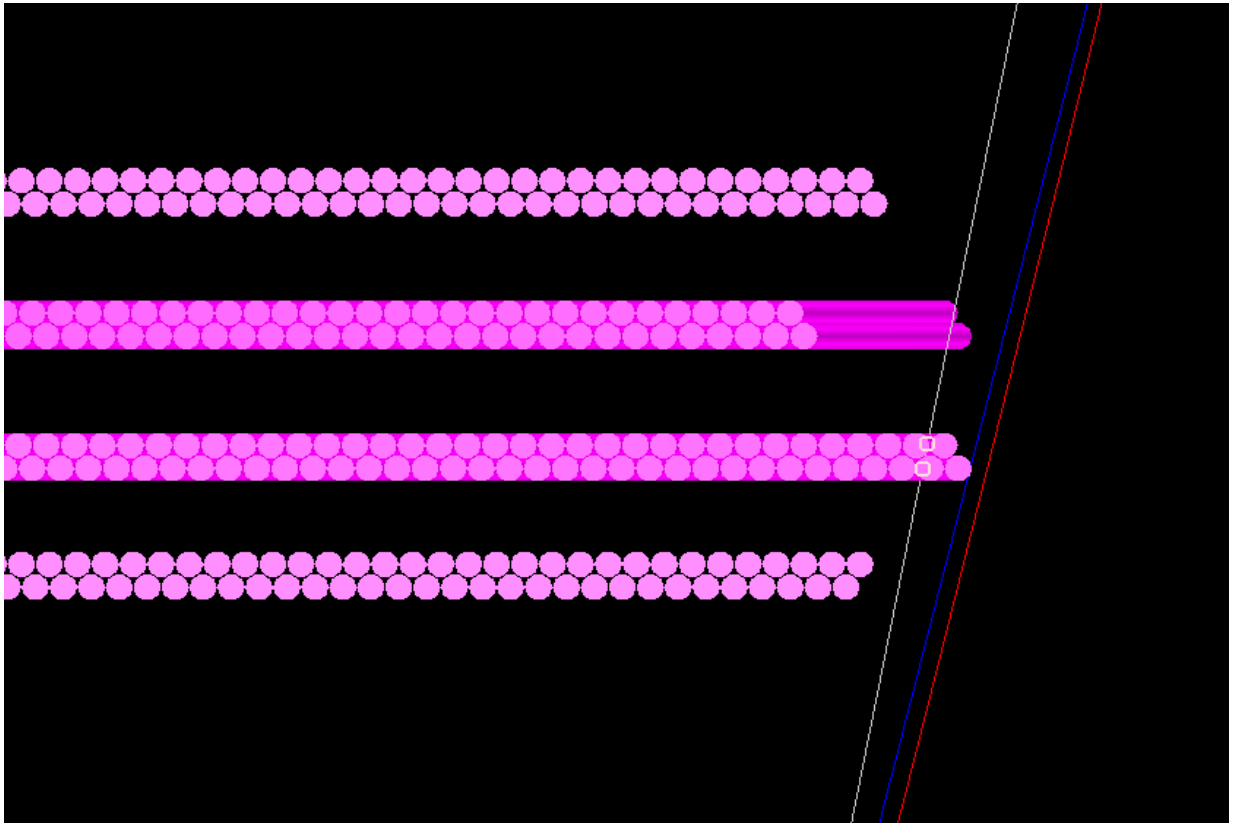
The algorithm can not distinguish between the different tracks

Quality analysis – events with many tracks



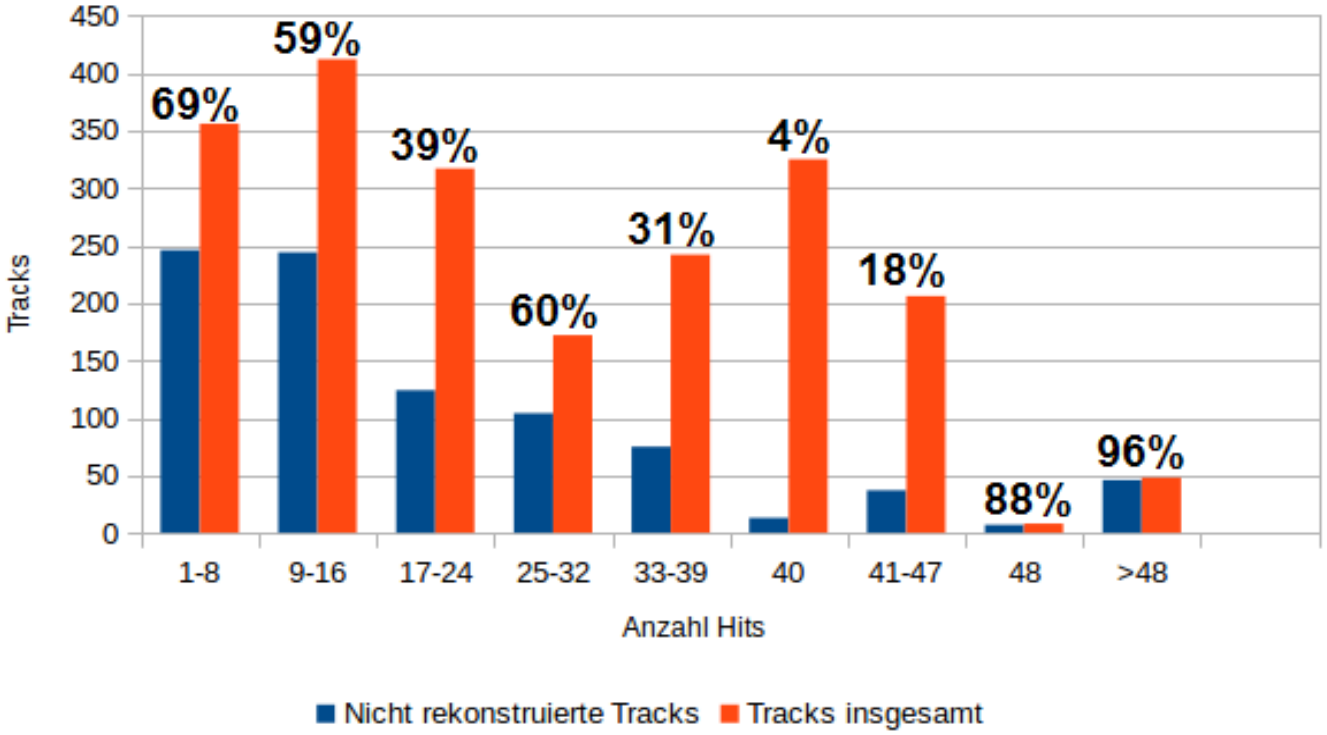
=> There is a correlation between many tracks in the event and not reconstructed tracks

Quality analysis – too little hits



The information of two hits is not enough for a correct reconstruction

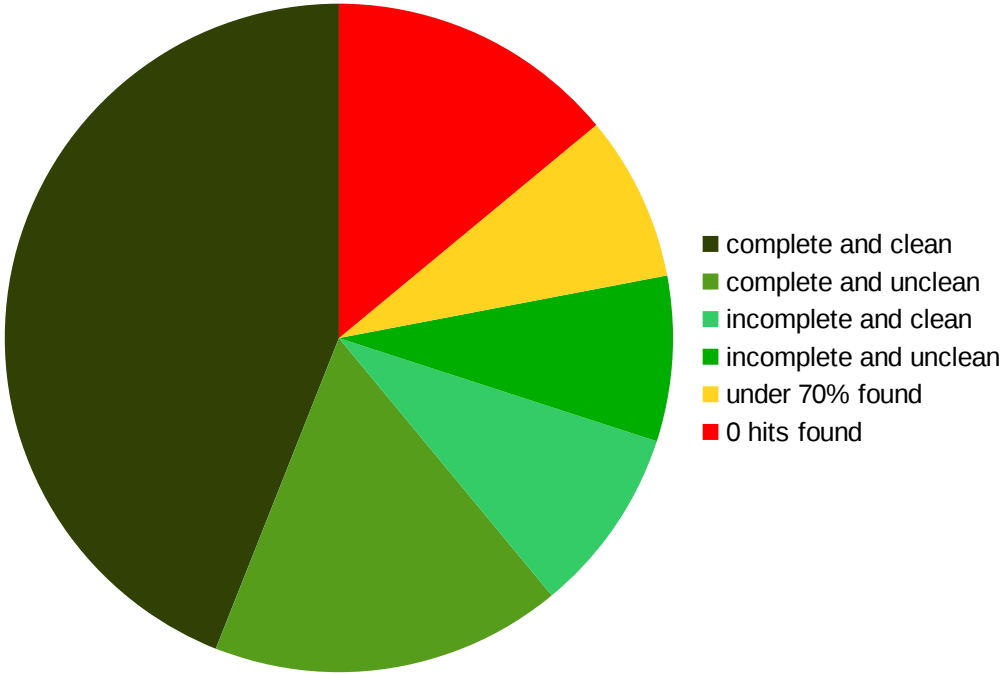
Quality analysis – too little hits



=> There is a correlation between not exactly 40 hits and bad reconstruction
40 hits = 2 hits per double layer * 4 double layers per station * 5 stations

FTS6 is currently not working in PandaRoot

Quality analysis



Analyze only the tracks with the following conditions:

No tracks with:
- momentum <0.3 GeV/c
- Hits < 8

No events with:
- Track count > 6

78% of the analyzed tracks could be reconstructed
44% of the analyzed tracks were complete and clean

Summary and Outlook

- First FTS-Trackfinder which is integrated in PandaRoot
- 78% of the important tracks could be found
- 44% of the important tracks can be reconstructed clean and complete
- Problems are
  - Rotating particles
  - Many tracks in an event
  - Too little hits

- Improve the way-follower:
  - Way-follower should remember the curvature of the track
  - Way-follower should use the curvature to find the right approximation
- Parallelization
  - PANDA has high requirements on runtime

Thank you for your attention