Parallel Algorithms for Online Trackfinding at $\bar{\text{P}}$ANDA

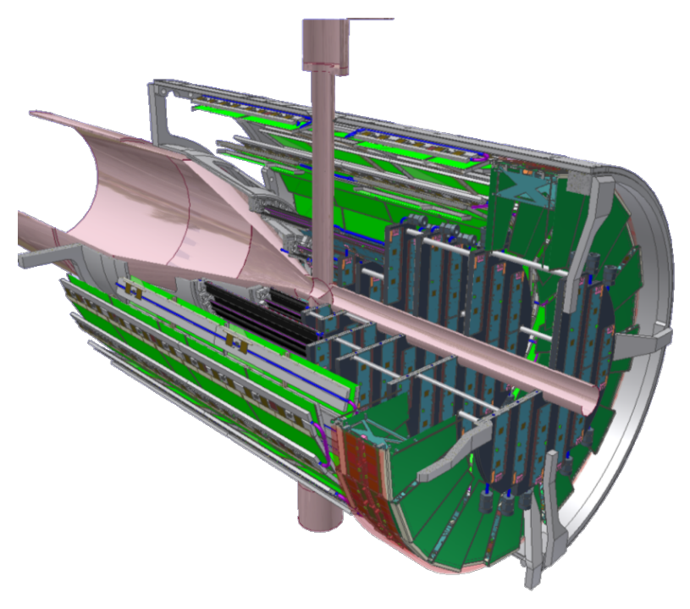
L. Bianchi[1], A. Herten[2], J. Ritman[1], T. Stockmanns[1]

[1] IKP, [2] JSC, Forschungszentrum Jülich GmbH, Germany

$\bar{\text{P}}$ANDA Experiment at FAIR

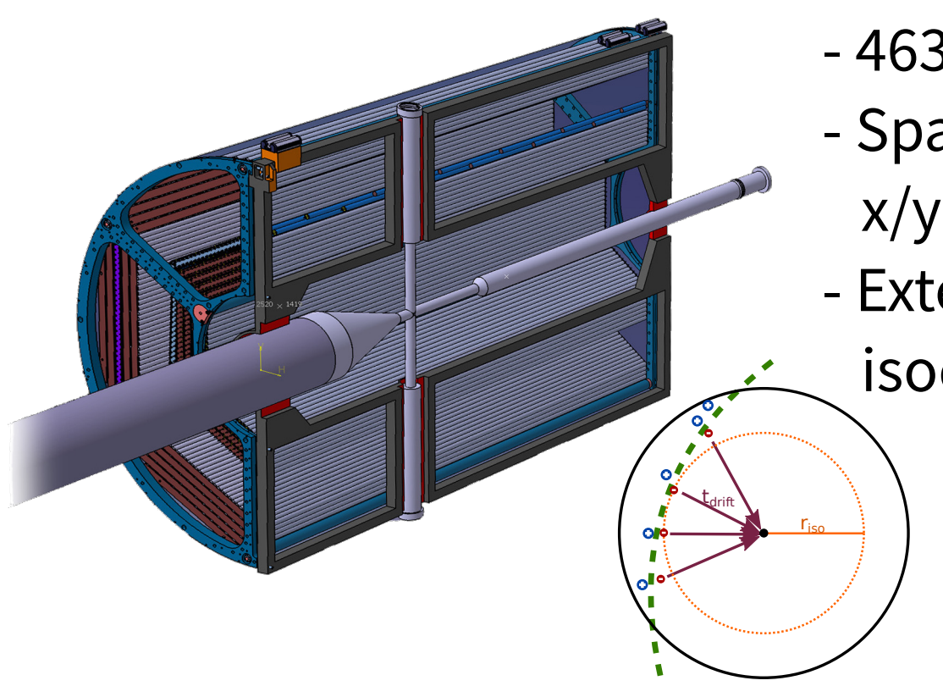
Central Tracking Detectors

Micro Vertex Detector (MVD)

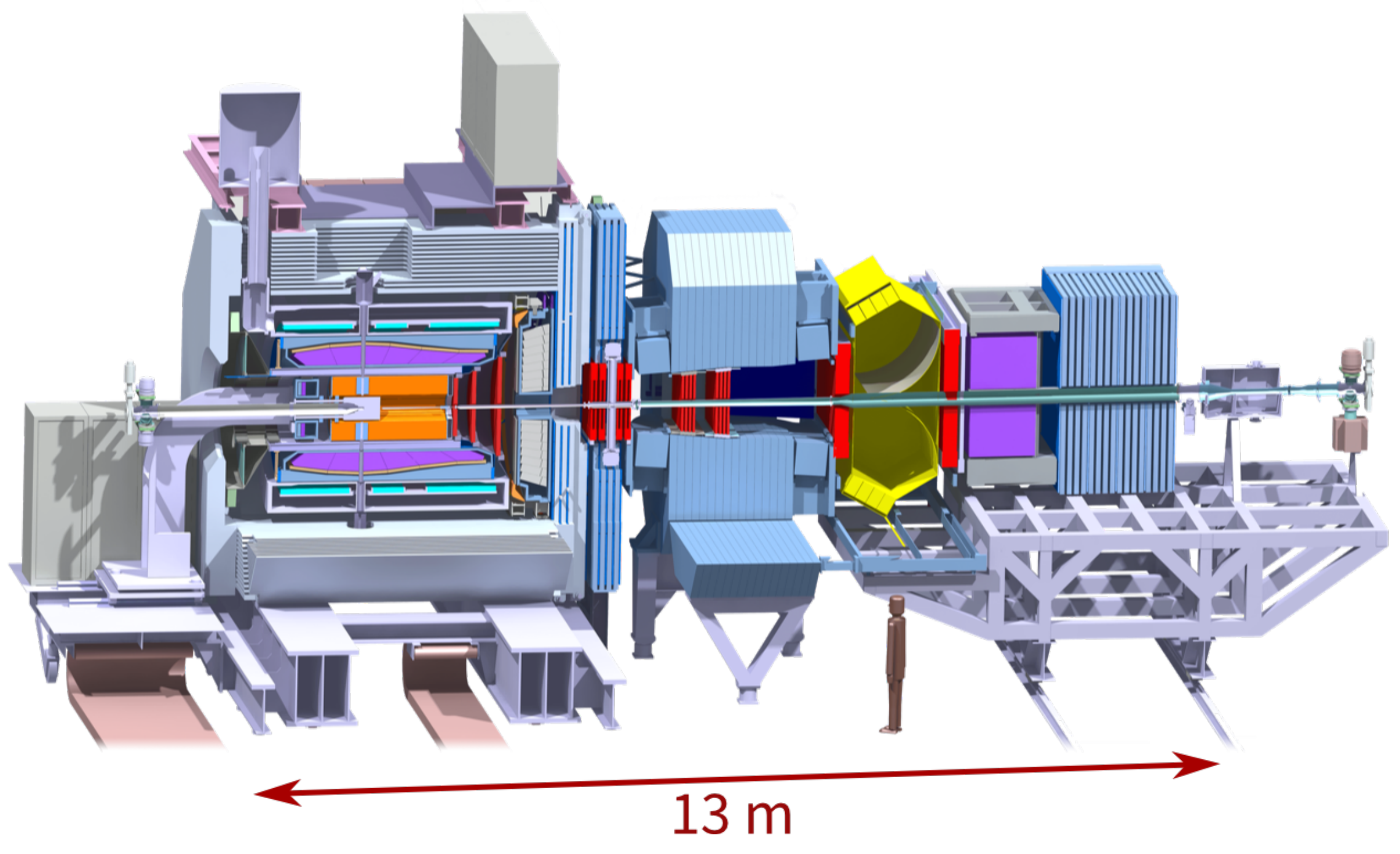


- Silicon detector with pixels (10.3M ch.) and double-sided strips (200k ch.)
- Vertex resolution ~100 µm
- Time resolution ~10 ns
- Point-like hits

Straw Tube Tracker (STT)



- 4636 straws (100 x 0.5 cm)
- Spatial resolution x/y ~150 µm, z < 3 mm
- Extended hits: isochrone radius from drift time

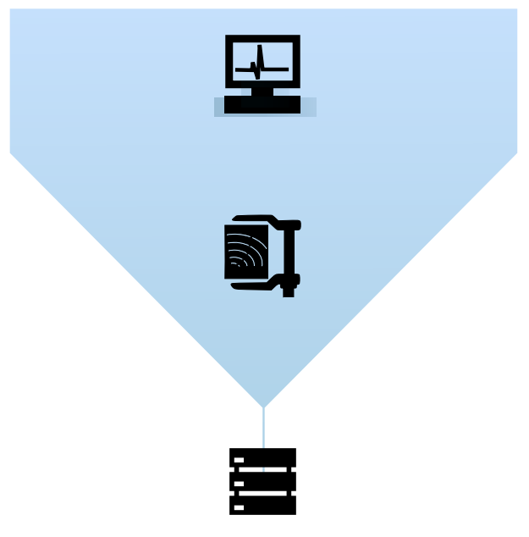


- Anti**P**roton **An**nihilation at **Da**rmstadt
- Fixed proton target
- 1.5 to 15 GeV/c antiproton beam
- Extensive research program in QCD physics

Data Acquisition Principles

- No hardware (L1) trigger
- Online event reconstruction and event selection
  ➔ **Online tracking**

Incoming data rate: up to **200 GB/s**



Offline storage: **3 PB/year**

➔ Data reduction factor: up to **1/1000**

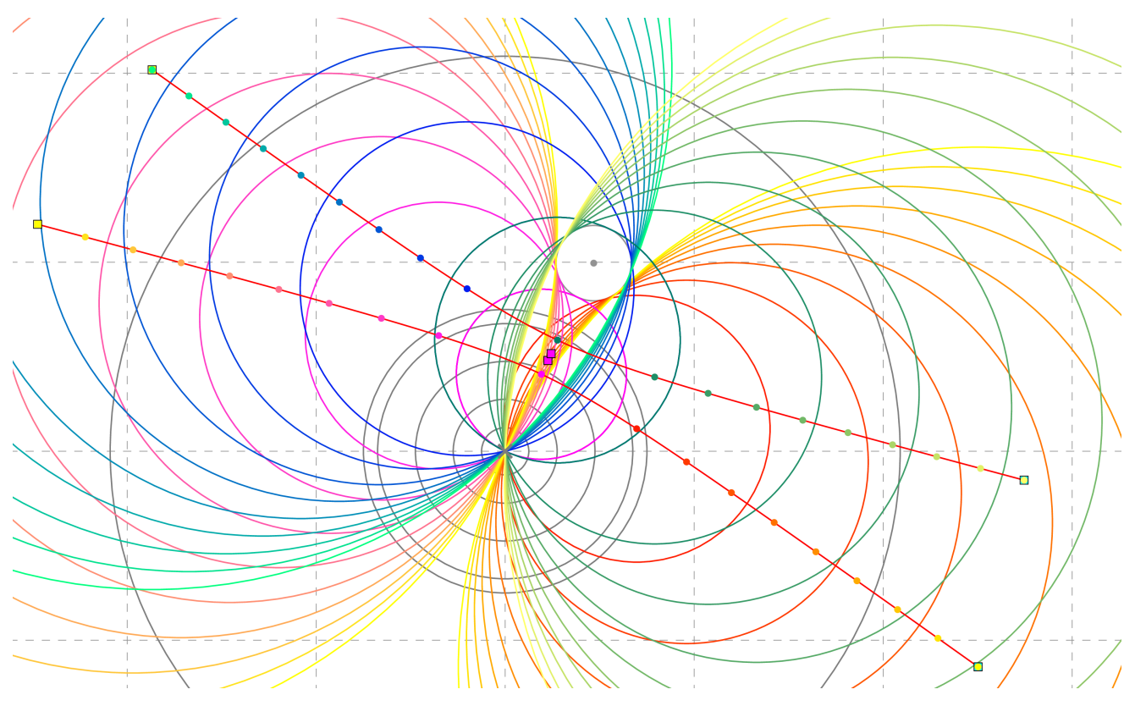
Circle Hough Algorithm

Application of Hough Transform to find particle tracks in a set of hits

1. For each hit, generate all possible tracks compatible with it
2. Collect track parameters for all hits
3. Most frequent sets of track parameters correspond to real tracks

(✓) Good trackfinding efficiency
(✓) Full hit information taken into account
(✓) Robustness
(✓) Applicable to all hits in central detectors
(✓) Intrinsic parallelism

Hough elements:
xy projections of primary tracks
➔ Circles in *xy* plane passing through IP and hit point
➔ 2D parameter space
➔ Coordinates of track centers

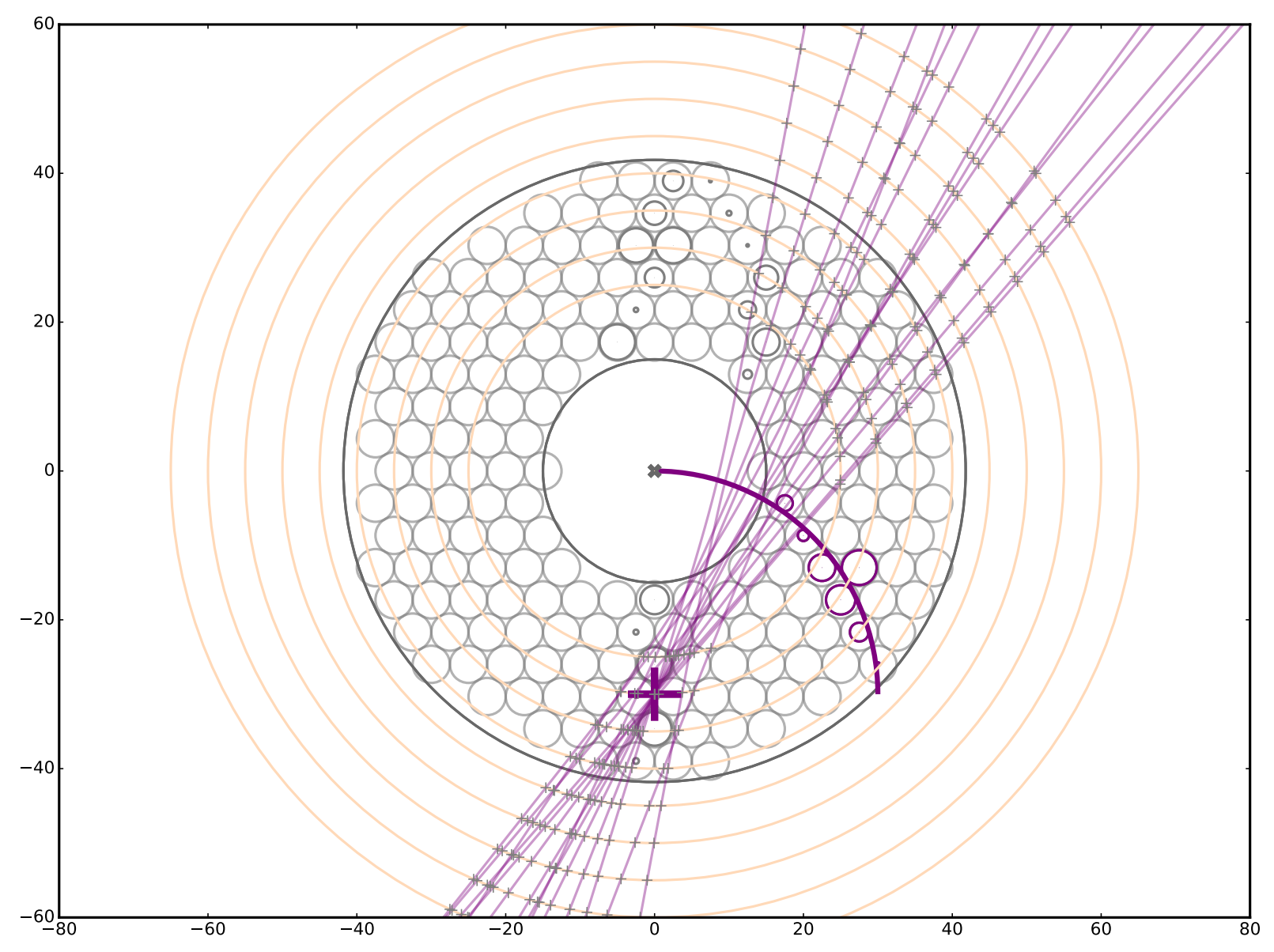


Geometric locus of track centers:
- Straight line (MVD hits)
- Hyperbola (STT hits)

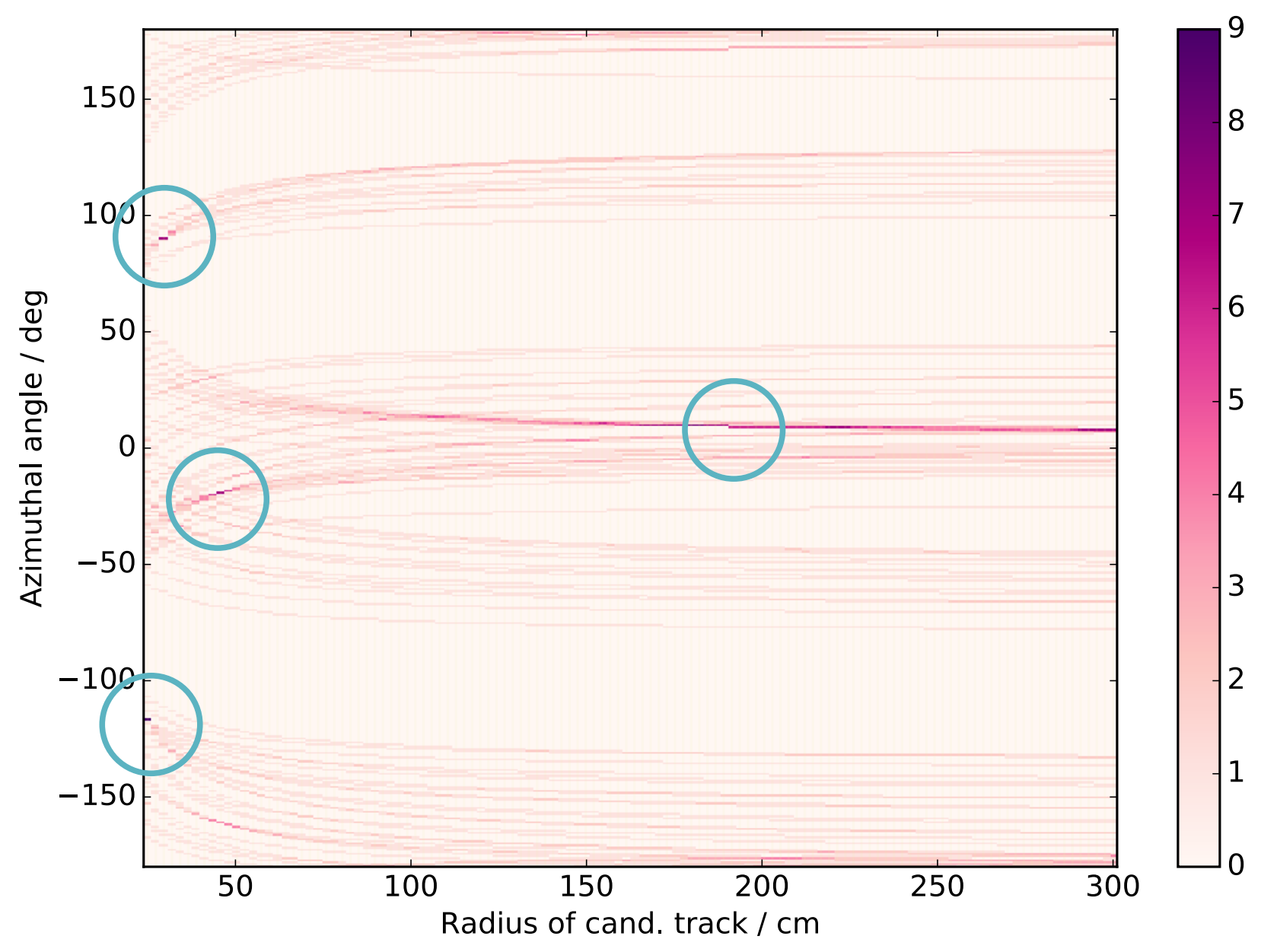
Collect elements:
- 2D Accumulator Array + peakfinding
- 2D clustering algorithms

Circle Hough for single hits

Hough element: track compatible with single hit

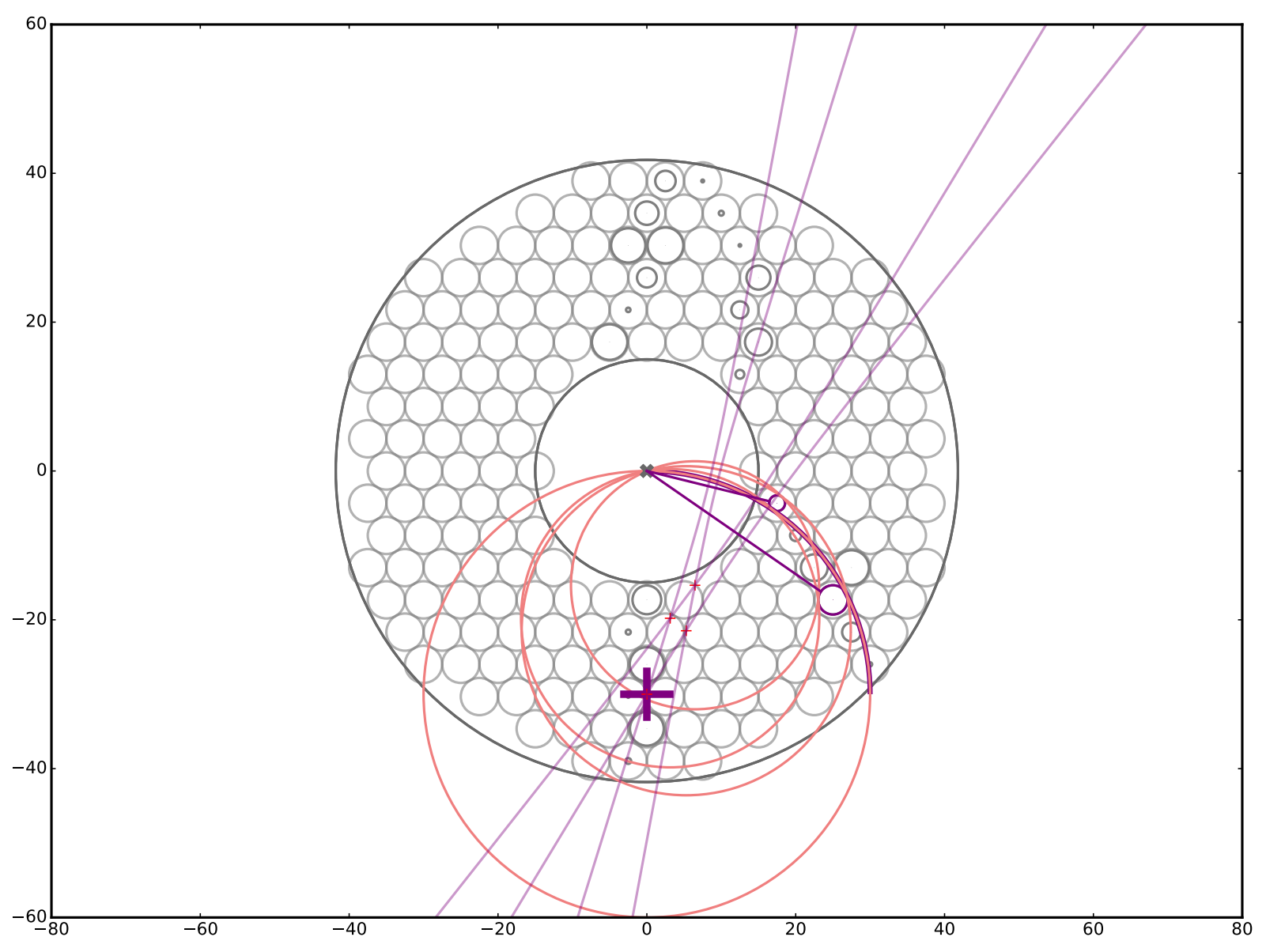


- Calculate track centers: (R, ϕ(R))
- Set of sampling values $\{R_0, \ldots, R_{\max}\}$ common for all hits
- 4 possible tracks per hit, *R*
- Accumulator Array (*R*, ϕ)
  - Match *R* bins and sampling values
  - Regular filling pattern
  - Parallel peakfinding
- Improve resolution: increase bin density recursively

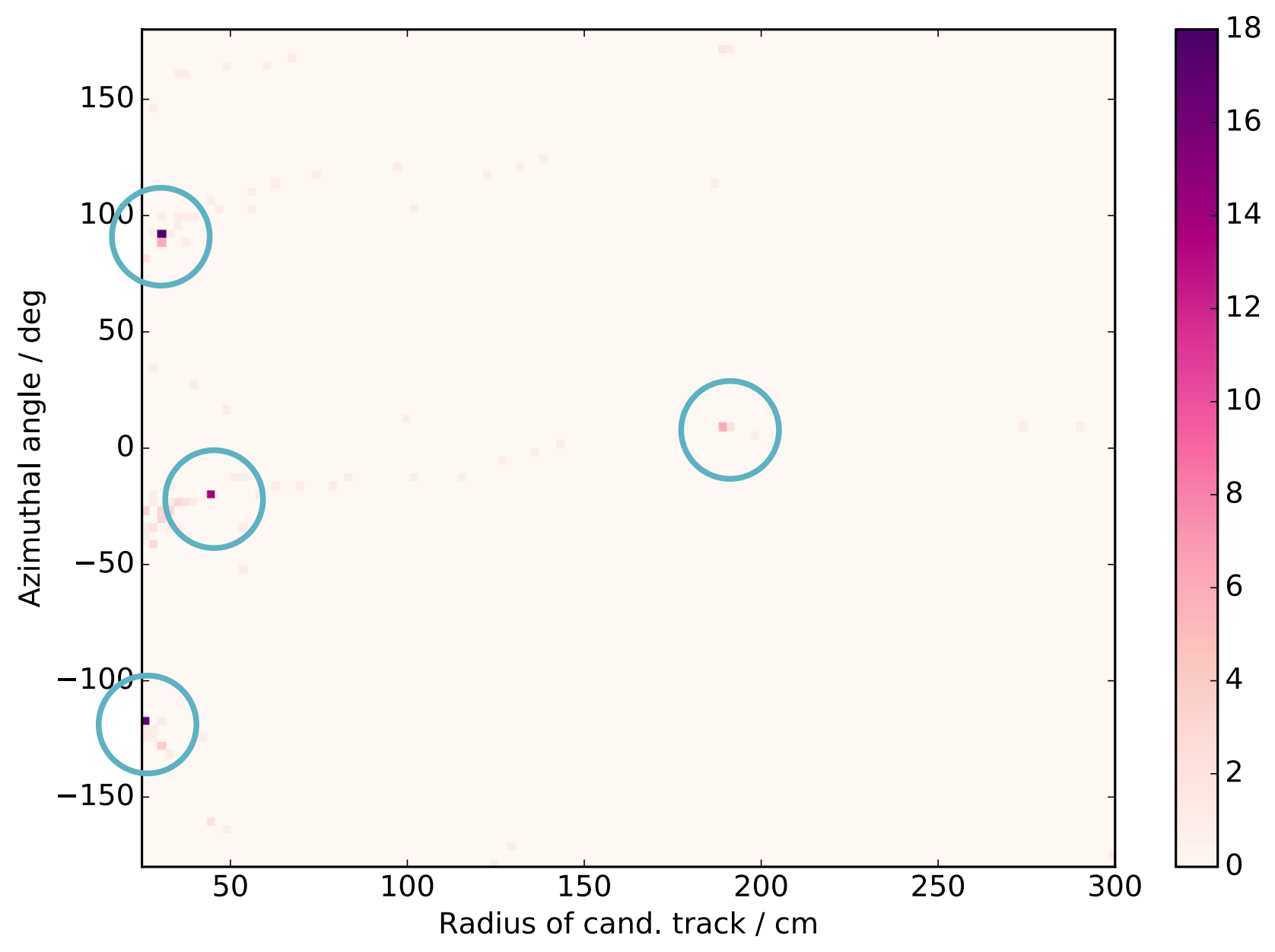


Circle Hough for hit pairs

Hough element: track compatible with two hits



- Reject spurious hit pairs
  simple criterion: Δϕ between hits in pair
- Calculate track centers:
  problem of Apollonius (PCC variant)
  "Find all circles passing through a point and tangent to two circles"
- 4 possible tracks per pair
- Different AA structure
  - Fewer unnecessary elements to collect
  - Simpler peakfinding algorithms

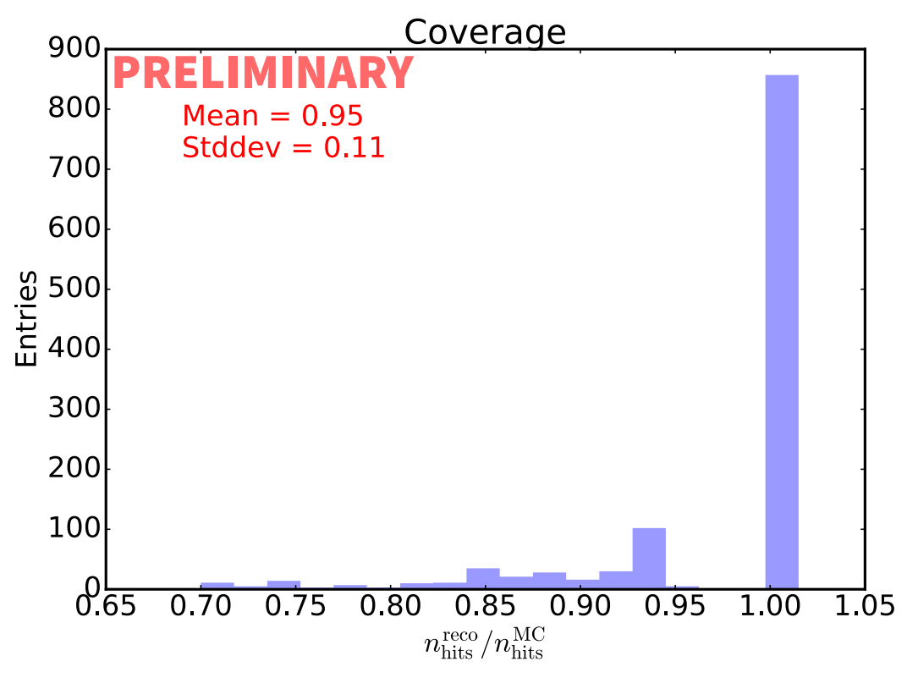


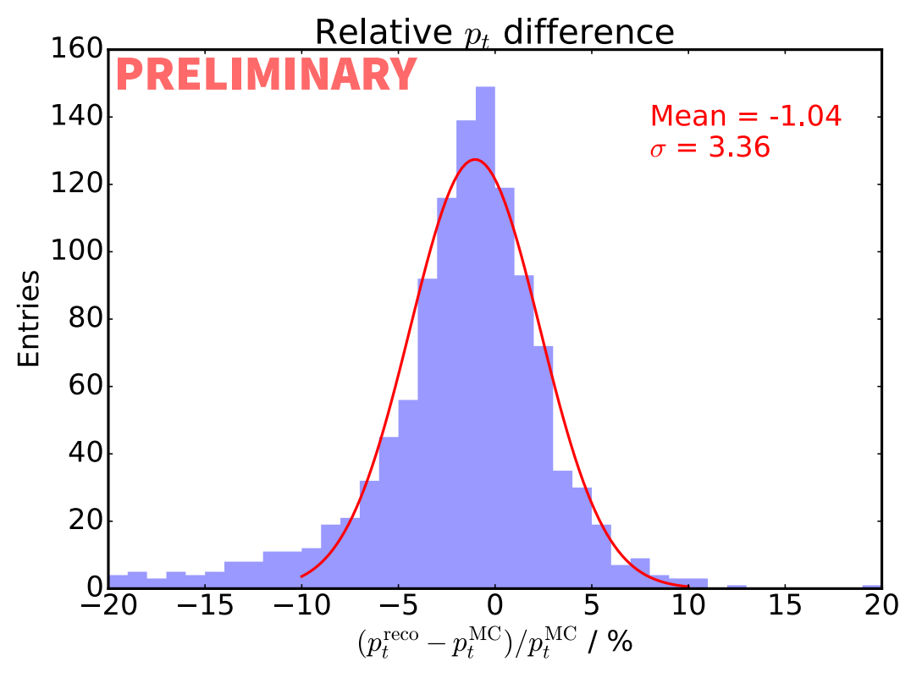
Performance

Primary tradeoff between:
- Reconstruction performance requirements
- Computing resources

Parameter optimization:
determine working point based on application

Online tracking:
- Trackfinding efficiency
- Hit coverage
- Initial p_t determination (without refitting)





Summary

- Novel variants of Circle Hough trackfinding algorithm to further extend parallelism
- Two versions in development
  - Single hits
  - Hit pairs
- Choice of parameter values determines working point for online and offline application

Outlook

- Integration into PandaRoot software framework
- Finalize code for measuring computing performance
  - CPU C++
  - CPU/GPU OpenACC
  - GPU Cuda C