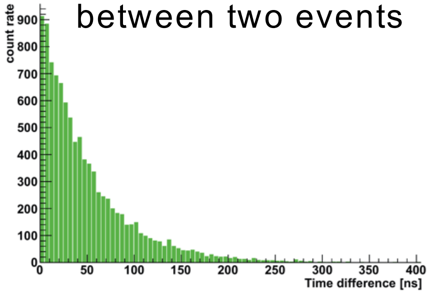
Algorithms for Online Tracking in PANDA

CHEP 2018

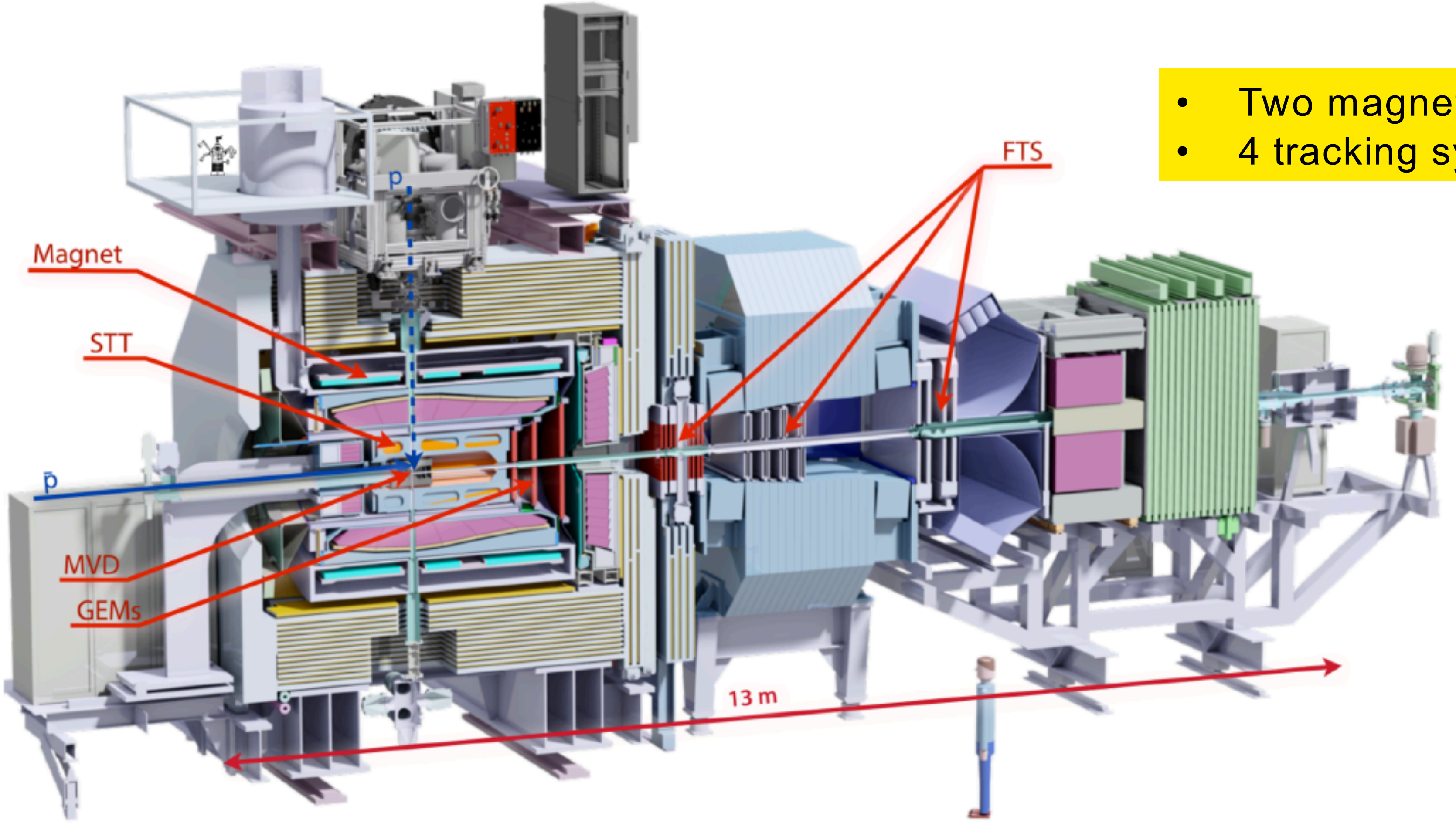
10.7.2018 TOBIAS STOCKMANNS

Boundary Conditions

Time difference between two events



- No time structure in beam
- Up to 20 MHz interaction rate
- Online tracking and event selection



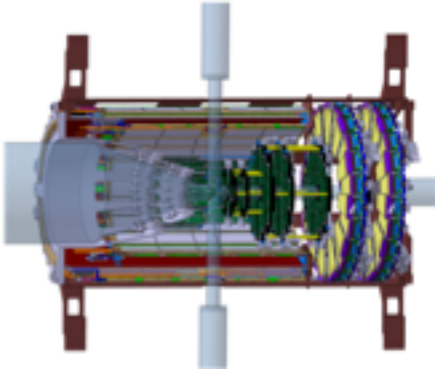
- Two magnets
- 4 tracking systems

- < 10 tracks / event
- Mainly p, π, K, e, μ

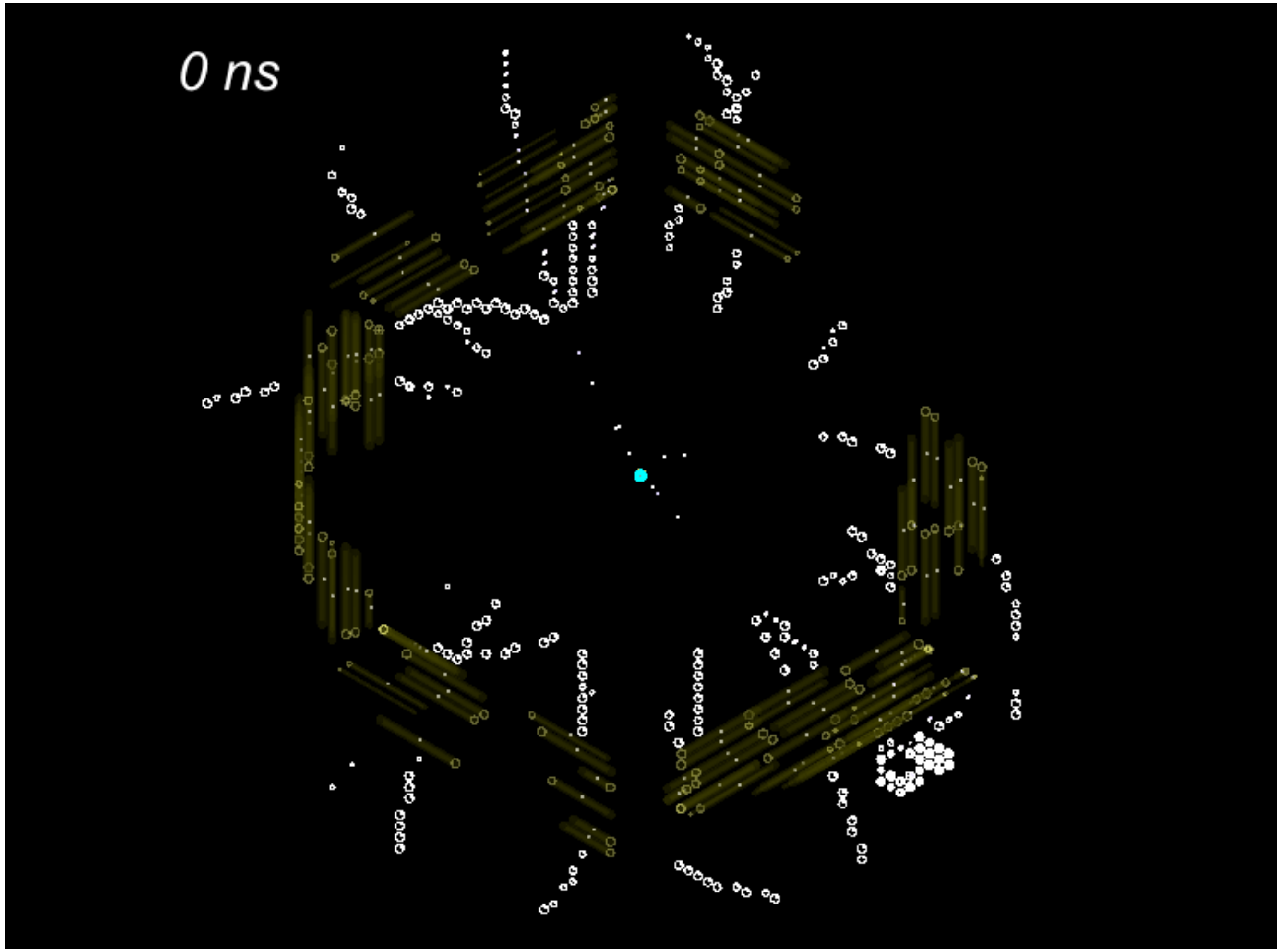
Simulated Events

MVD

- 4 barrel layers
- 6 disks
- Mixed pixels / strips
- 3D space points
- < 30 µm point resolution
- < 10 ns time resolution



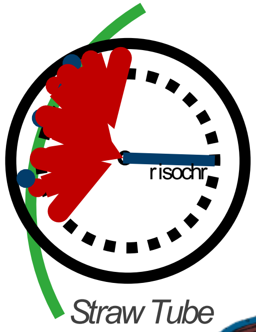
MVD

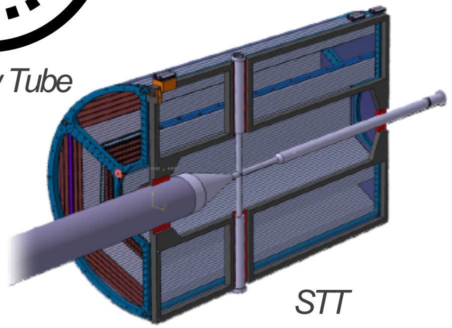


Time series of MVD and STT detector signals in PANDA (view along beam axis)

STT

- ~ 4000 straws
- Dense packaging
- < 150 µm resolution
- Isochrones
- 250 ns drift time
- 2D space point
- No start time

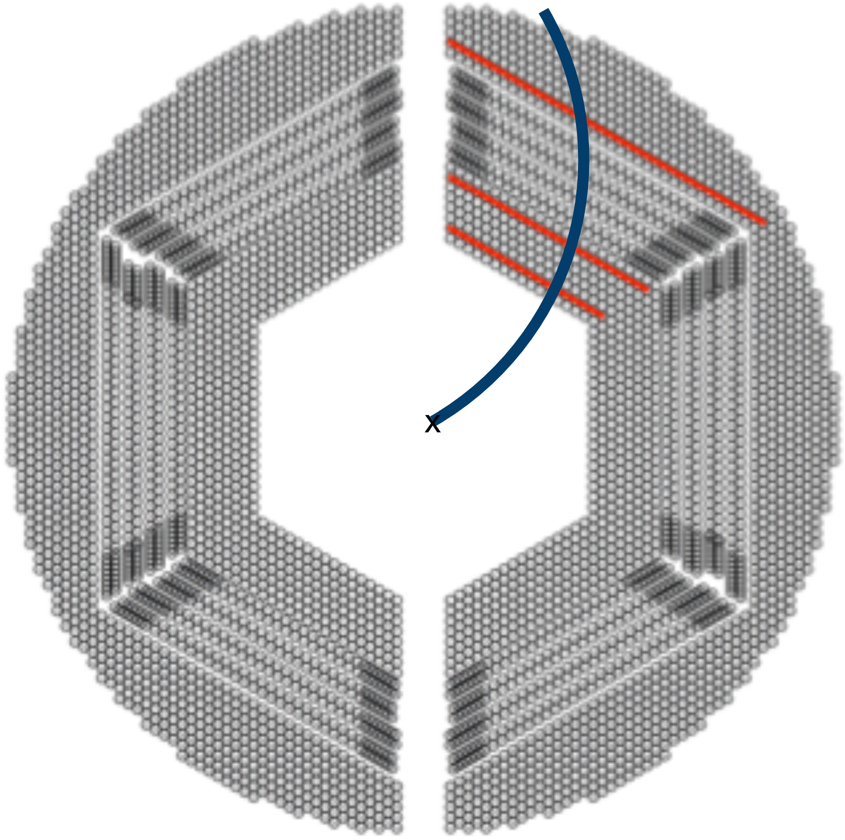




STT

Triplet Finder

- Find 2 out of 3 combinations of hits in pivot layers and primary vertex
- Calculate circle
- Add hits in other layers by distance to circle
- Straight line fit for z-component
- Select track candidates by number of hits

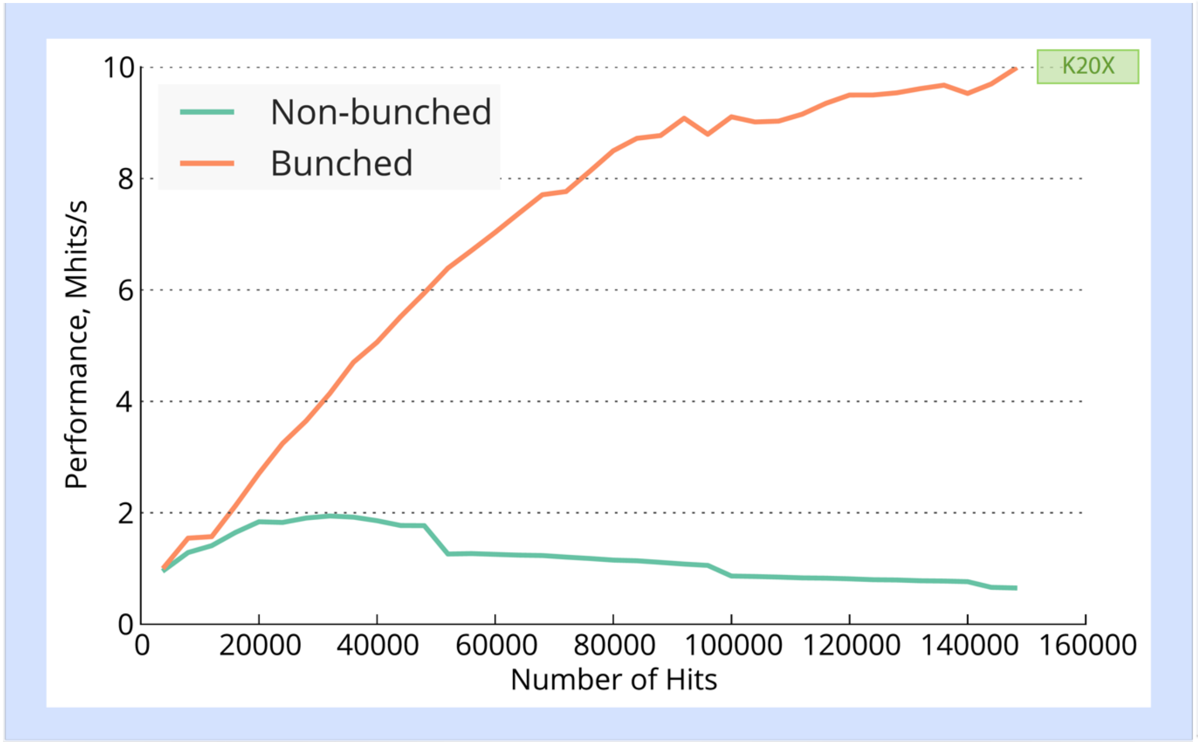


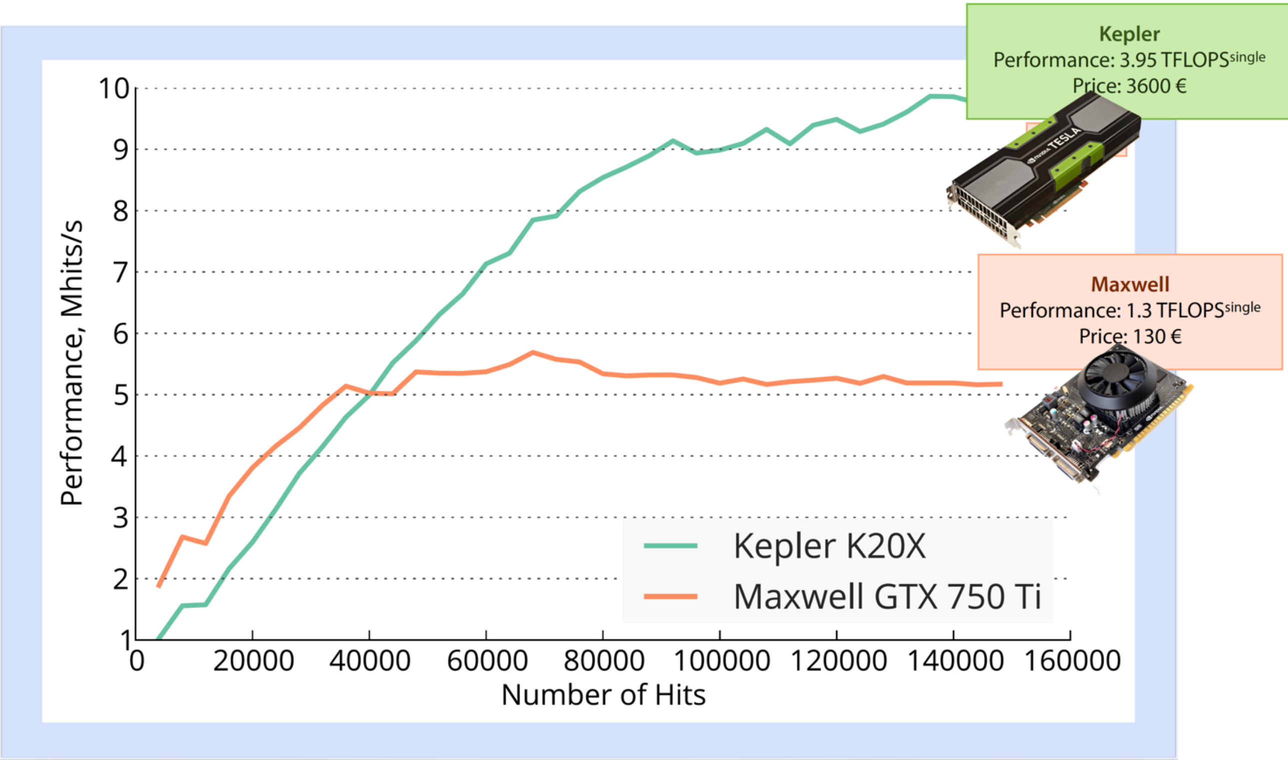
+ Simple algorithm
+ No fitting needed
- No isochrone information
- Limited on primary tracks

- Optimized for GPUs together with Nvidia Application Lab

Marius Mertens(FZJ), Andrew Adinetz(FZJ), Andreas Herten(FZJ)

Triplet Finder

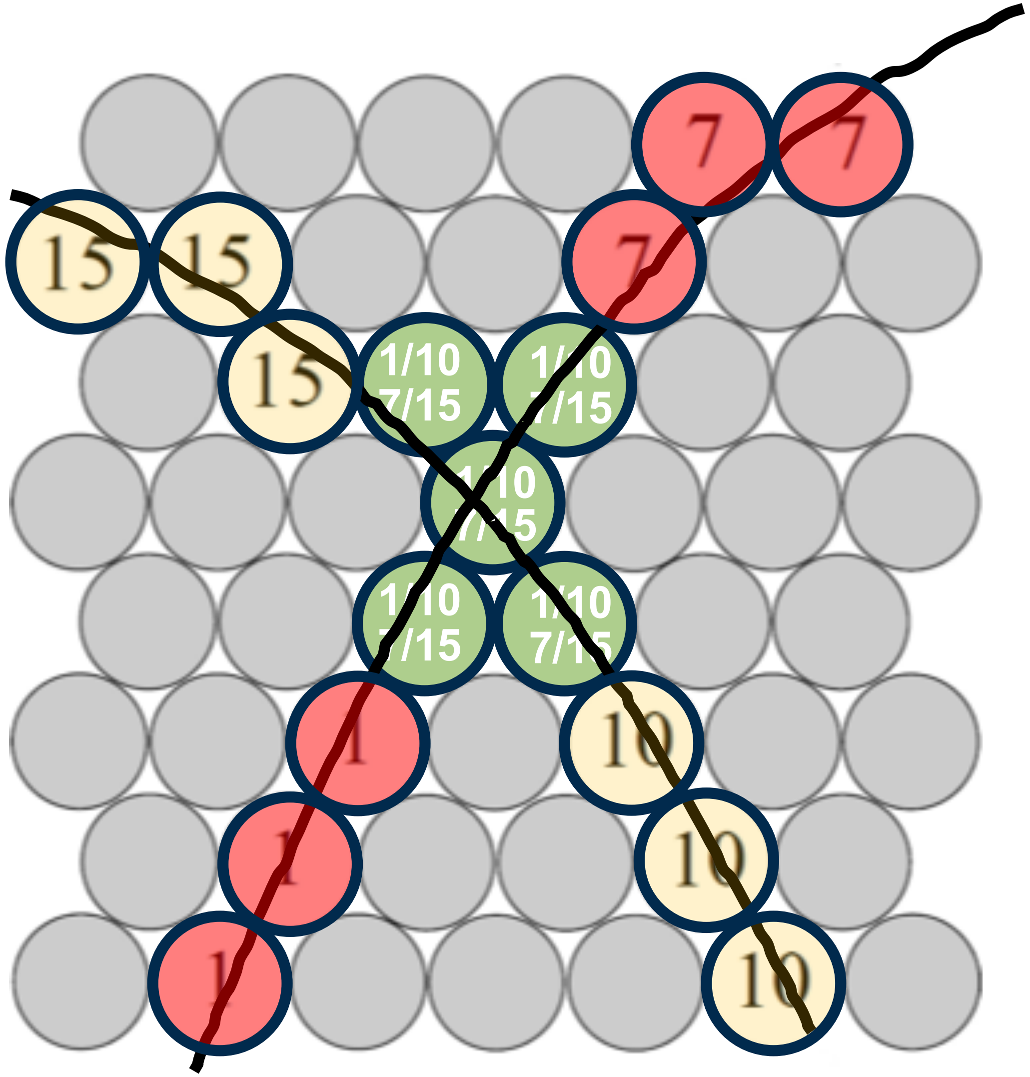




- Non-bunched: packages of single events send
- Bunched: Fixed time bunch with many events

Cell Track Finder

- Find (unambiguous) neighbours with hits → generates tracklets
- Find connections between tracklets
- Fit connected tracklets and take best matching one (Circle fit via Riemann Surface)
- Add the ambiguous hits
- Correct for isochrones
- Straight line fit for z-component



\+ All hits processed at once

\+ Primary and secondary tracks

\+ Generall clustering algorithm

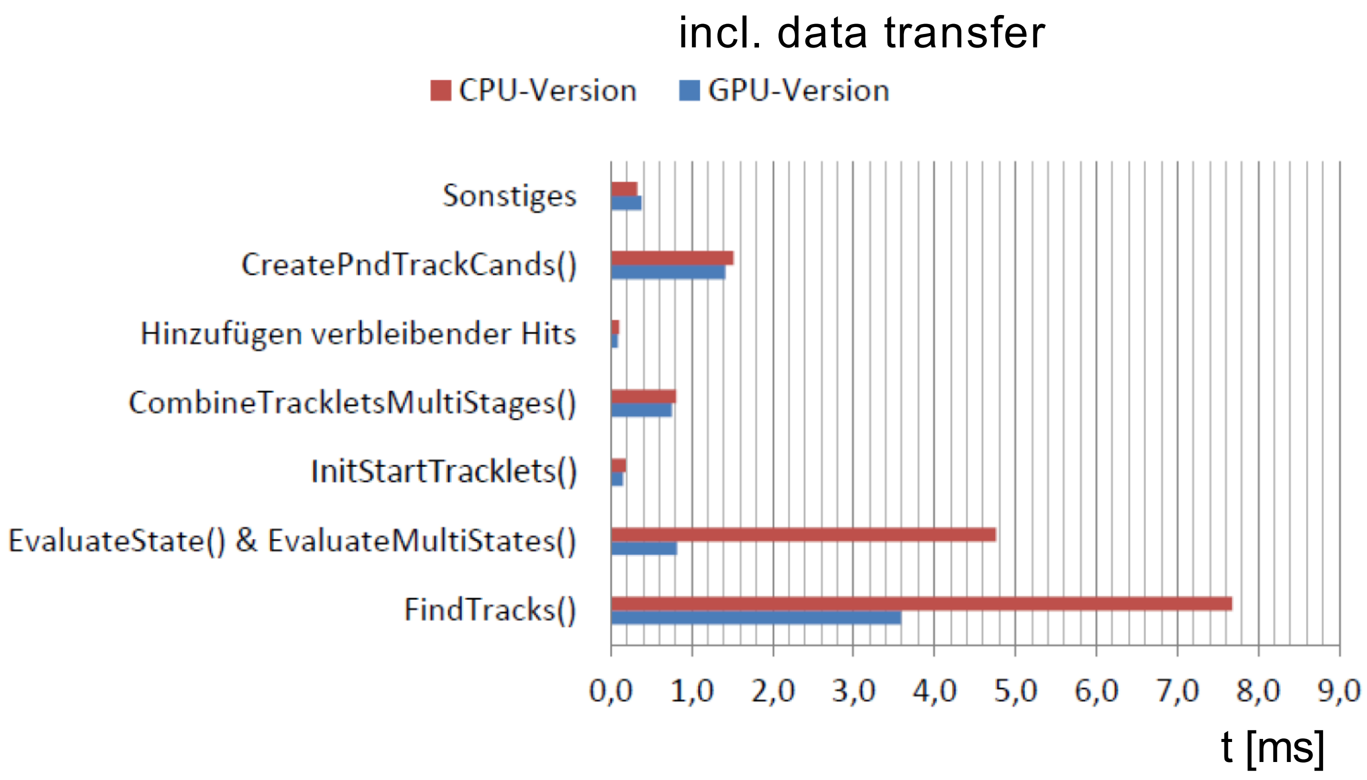
o Isochrone information in an additional stage

\- Only suitable for STT

- Optimized for GPU

Cell Track Finder

Processing time per event

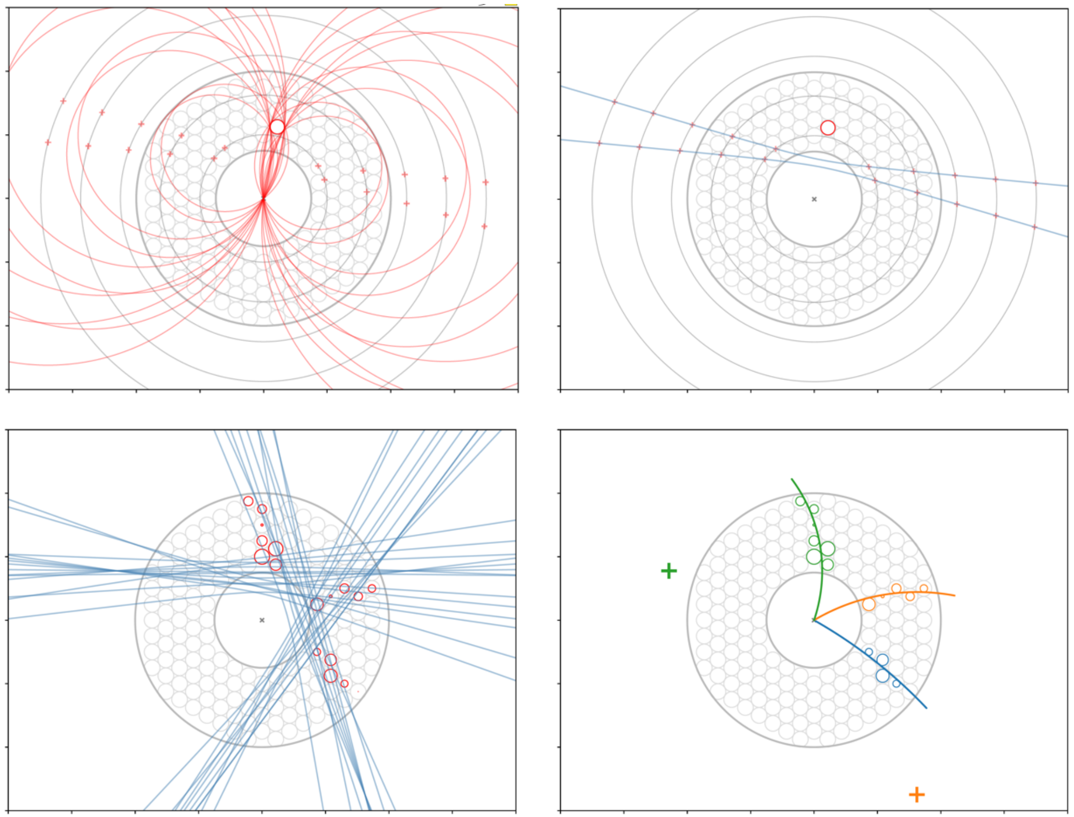


Process neighbourhood relations
(EvaluateState & EvaluateMultiStates)

Version	Runtime per event	Ratio of CPU Time
CPU	4.56 ms	100 %
GPU (algorithm)	0.575 ms	12.08 %
GPU (+ event)	0.045 ms	0.94 %

Circle Hough

- Calculate for all hits all circles through origin which are tangent to isochrone
- Fill x, y coordinate of circle center into histogram
- Find peaks
- Straight line fit for z-component



+ All hits in parallel

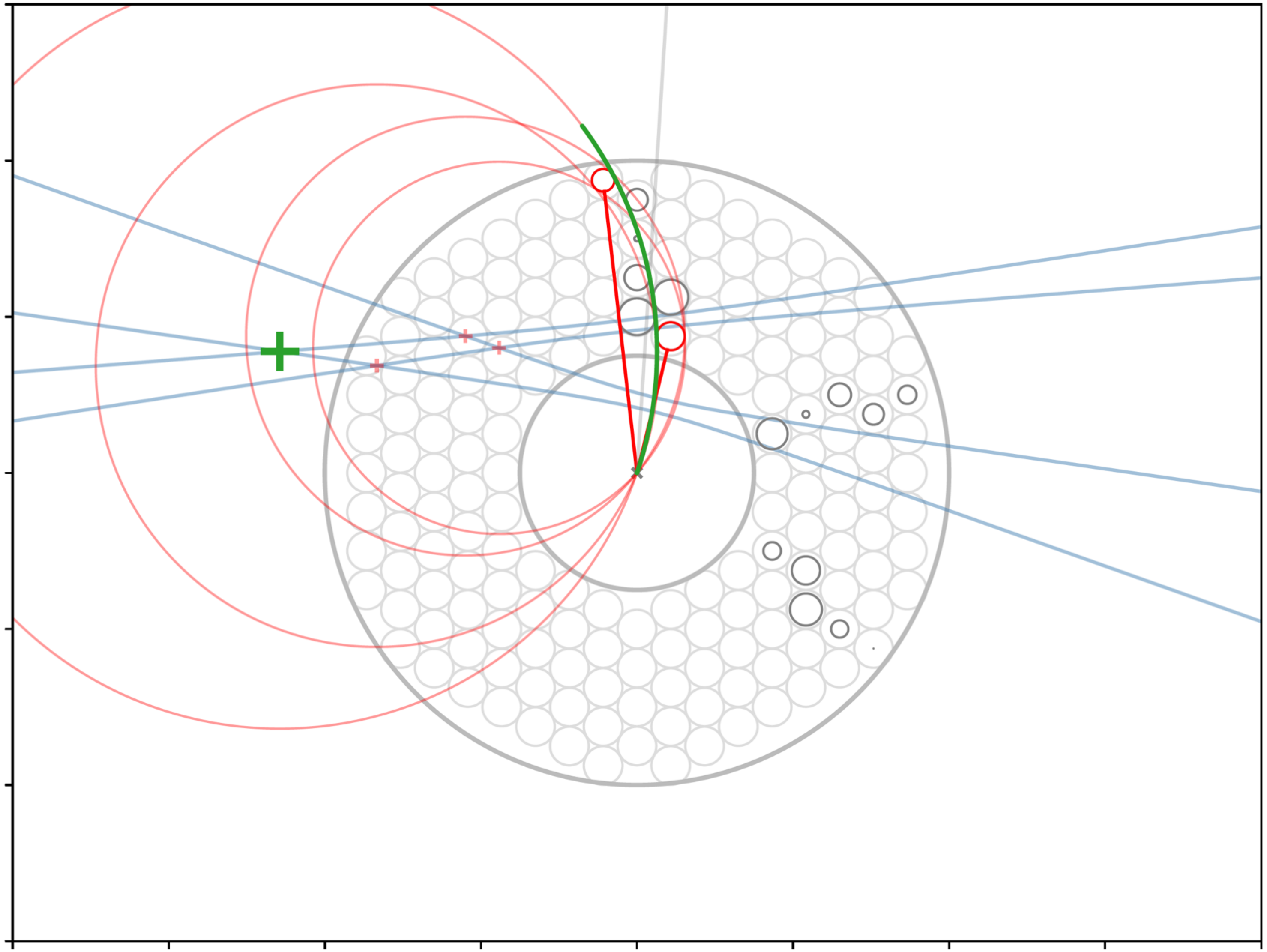
+ Isochrone information naturally in

+ Works for STT and MVD

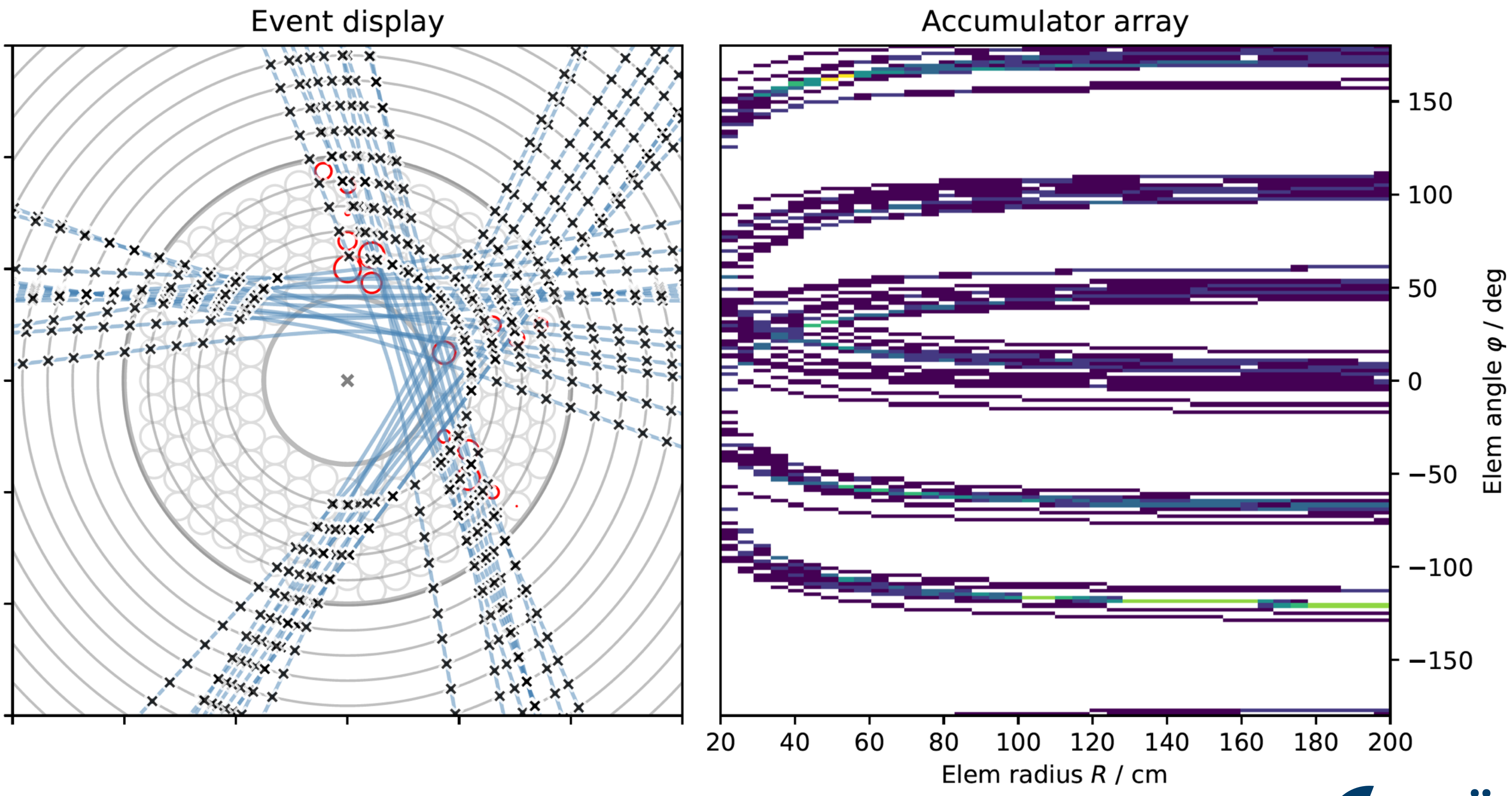
- Limited on primary tracks

Andreas Herten (FZJ),
Ludovico Bianchi (FZJ)

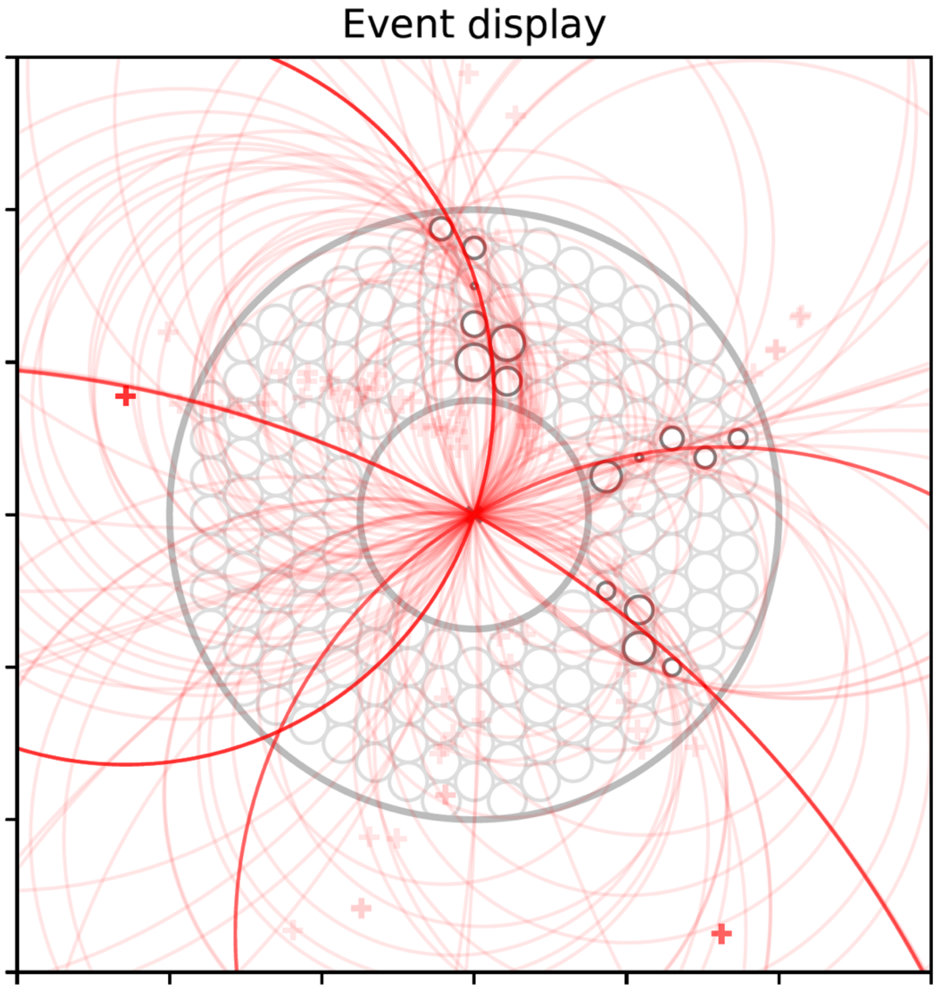
Pair Circle Hough

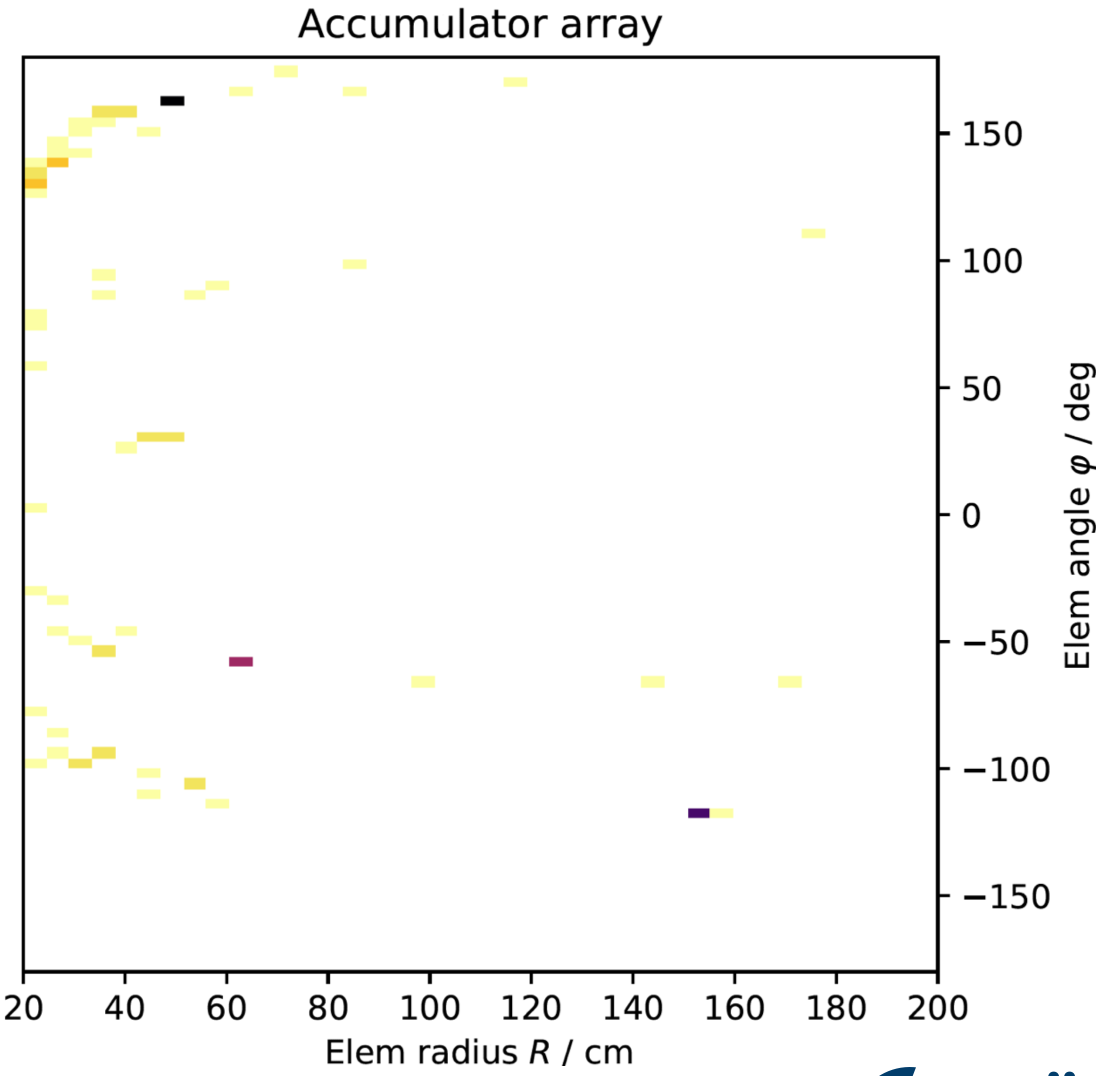


Circle Hough



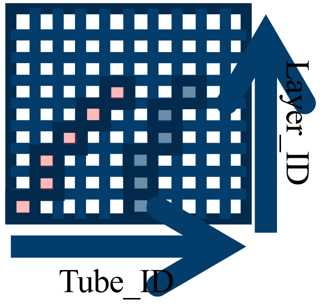
Pair Circle Hough





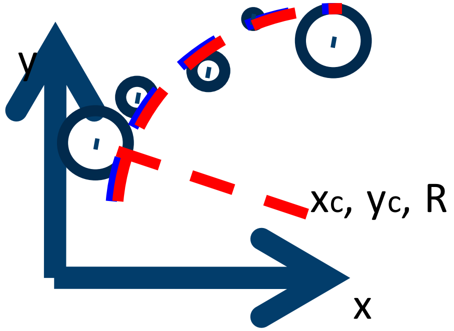
Track Finding on FPGA

- Road Finding



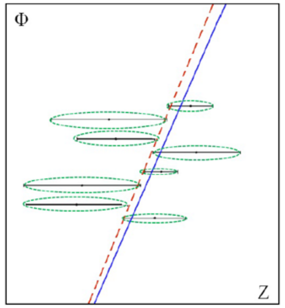
- Circle Fitting

$$E^2 = \sum(x_i^2 + y_i^2 + a\, x_i + b\, y_i + c)^2 (1/d_i)^2$$



- Straight Line Fitting

$$E^2 = \sum(\Phi_i + k z_i + \Phi_0)^2 (1/d_i)^2$$



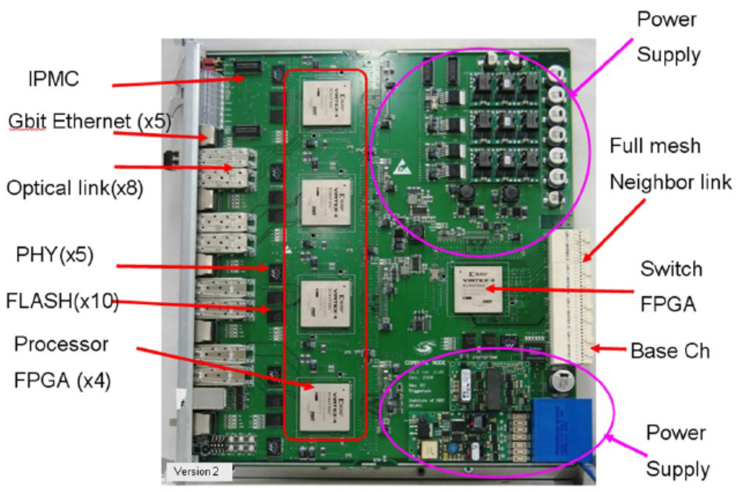
+ Hits processed sequentially
+ Isochrone information used
+ T0 extraction possible

- Only suitable for STT

- Optimized for FPGA

Yutie Liang (JLU Giessen)

Performance



- Efficiency > 90 %
- σ_{pt} : ~ 3.2% σ_{pz} : ~ 4.2% .
- 7 µs/event (6 tracks)
- Tracking strategy in case of missing T0.

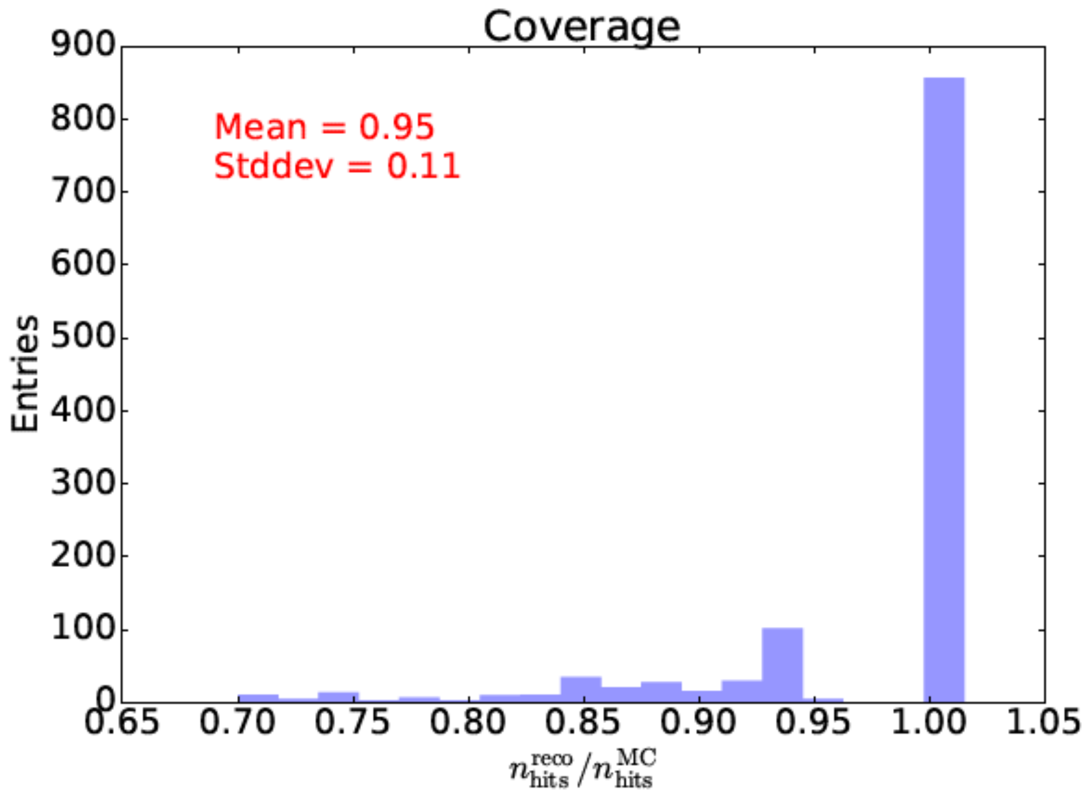
Summary

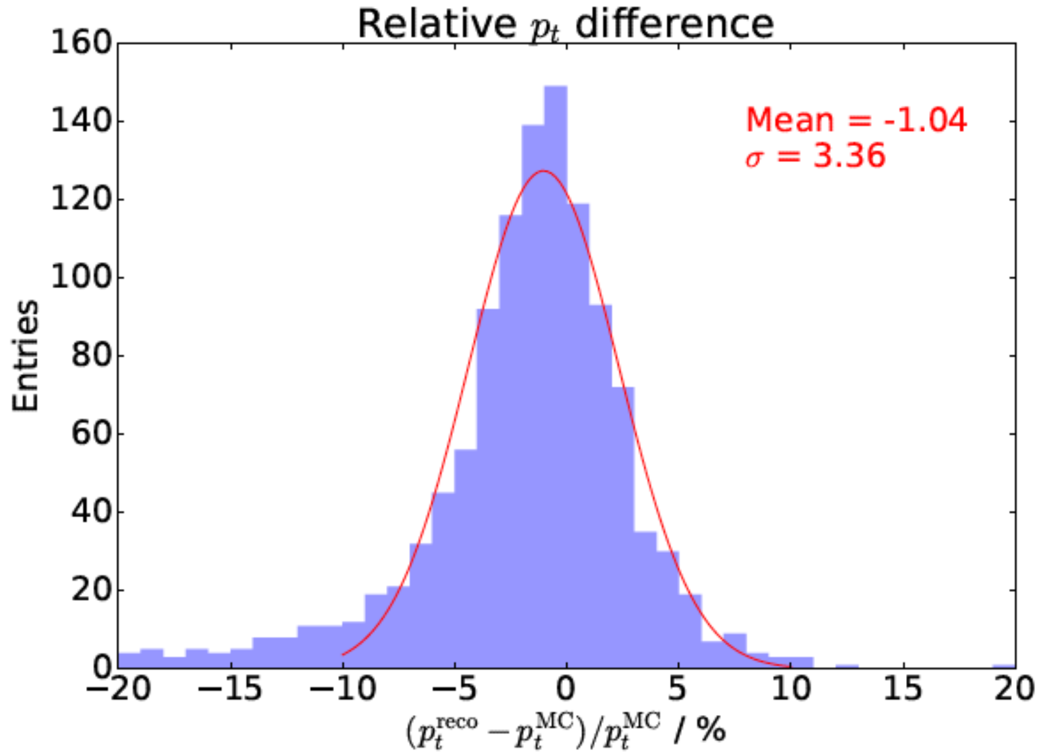
- Tracking in PANDA challenging
- Several different algorithms exists in different states of maturity
- No best candidate obvious
- Common test environment finished

- Next steps:
  - Finalize the algorithms
  - Test all on common data sample
  - Chain or combine algorithms
  - Machine Learning as alternative?

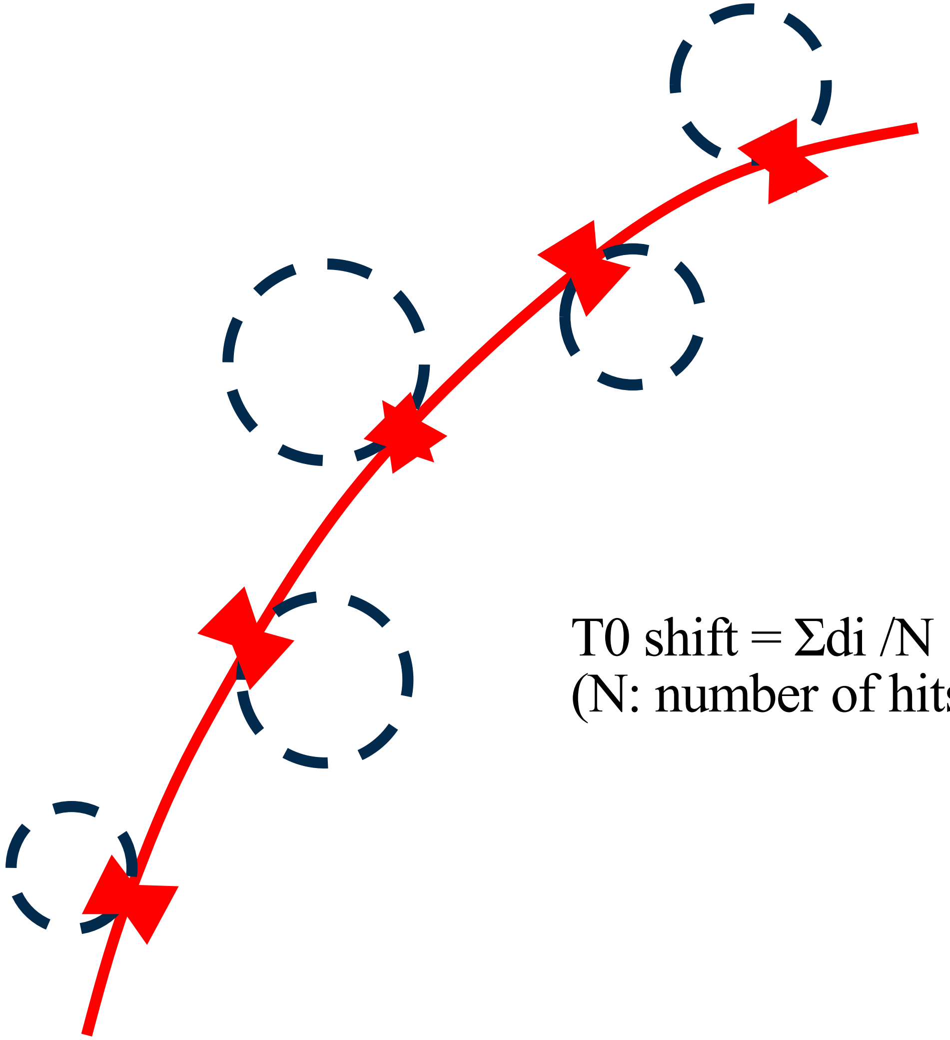
Backup Slides

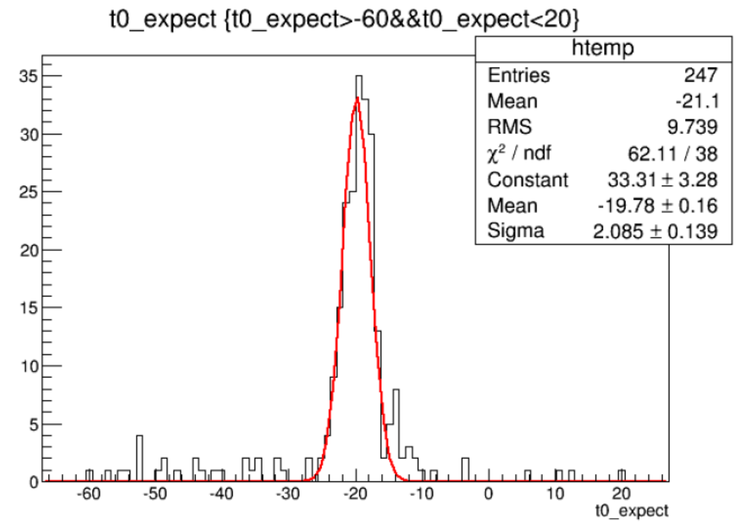
Circle Hough





T0 determination

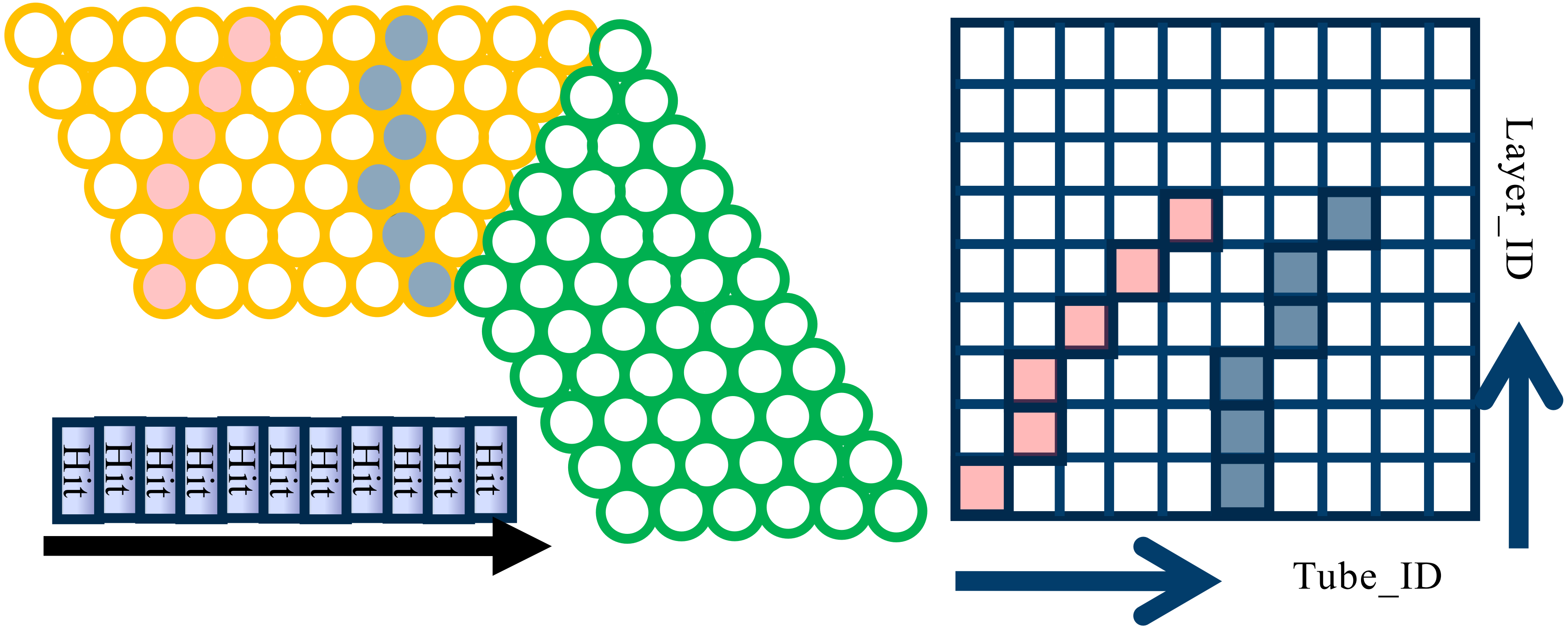




T0 shift = Σdi /N / const
(N: number of hits di: signed distance of circle to track)

T0 shifted by:	Extracted T0:
-50 ns:	-47.0±4.0 ns
-20 ns:	-19.8±2.1 ns
10 ns:	9.4±2.5 ns
20 ns:	19.0±2.3 ns

Tracking Algorithm -- Road Finding



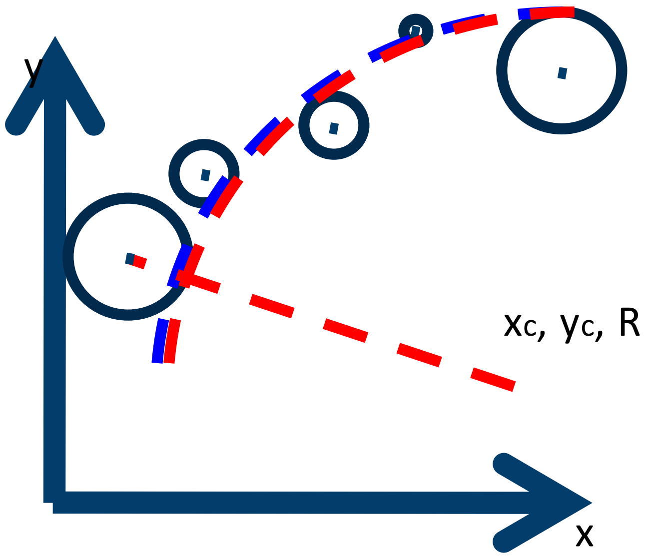
Hit: Seg_ID (3 bits) + LayerID (5 bits) + Tube_ID (6 bits) + Arrival time

1: Start from inner layer

2: Attach neighbour hit to tracklet layer by layer

- ✓ Boundary between two segments.
- ✓ Number of neighbor:

  4 in axial layer; 6 in stereo layer



Known : x_i , y_i , d_i

Question: To determine a circle,

$$x^2 + y^2 + ax + by + c = 0$$

Method: Minimize the equation

$$E^2 = \sum(x_i^2 + y_i^2 + a\, x_i + b\, y_i + c)^2 (1/d_i)^2$$

1) Circle para.

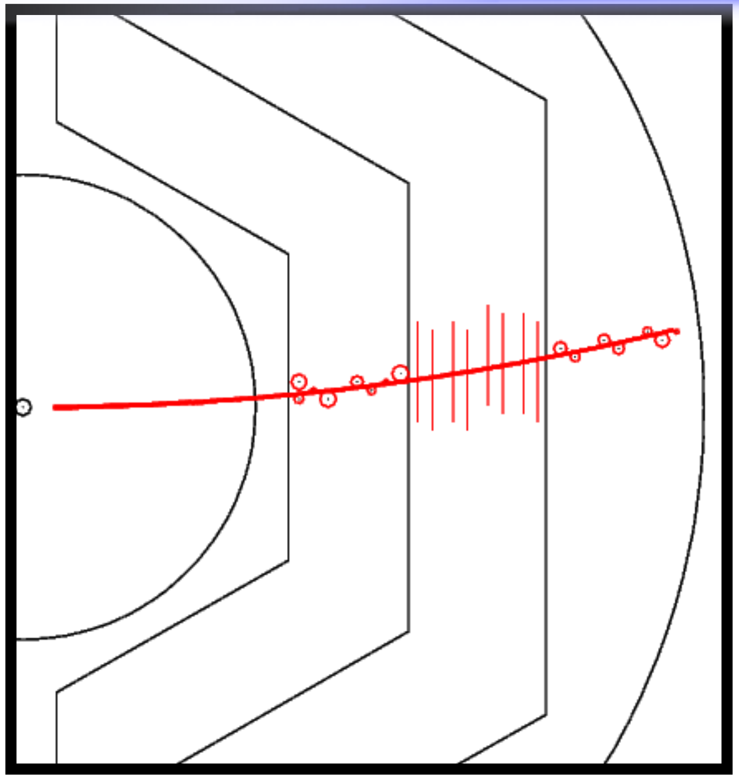
$$\begin{pmatrix} S_{xx} & S_{xy} & S_x \\ S_{xy} & S_{yy} & S_y \\ S_x & S_y & N \end{pmatrix} \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} -S_{xxx} - S_{xyy} \\ -S_{xxy} - S_{yyy} \\ -S_{xx} - S_{yy} \end{pmatrix}$$

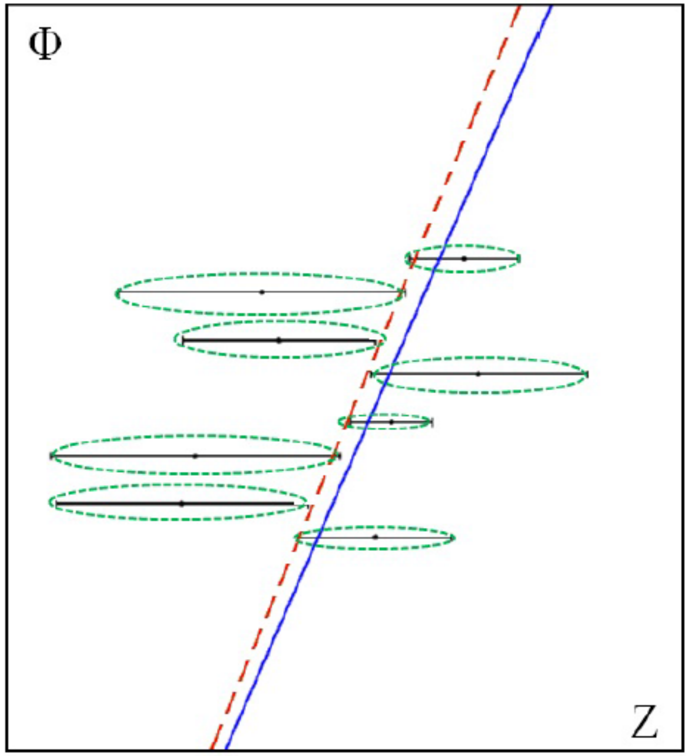
$S_x = \sum x_i$ …
$S_{xx} = \sum x_i x_i$ …
$S_{xxx} = \sum x_i x_i x_i$ …

2) Track quality.

$$\chi^2 = 1/n \times \sum_i \frac{(x_i^2 + ax_i + y_i^2 + by_i)/2r^2}{d_i^2}$$

Pz reconstruction





Known : z_i , Φ_i , d_i

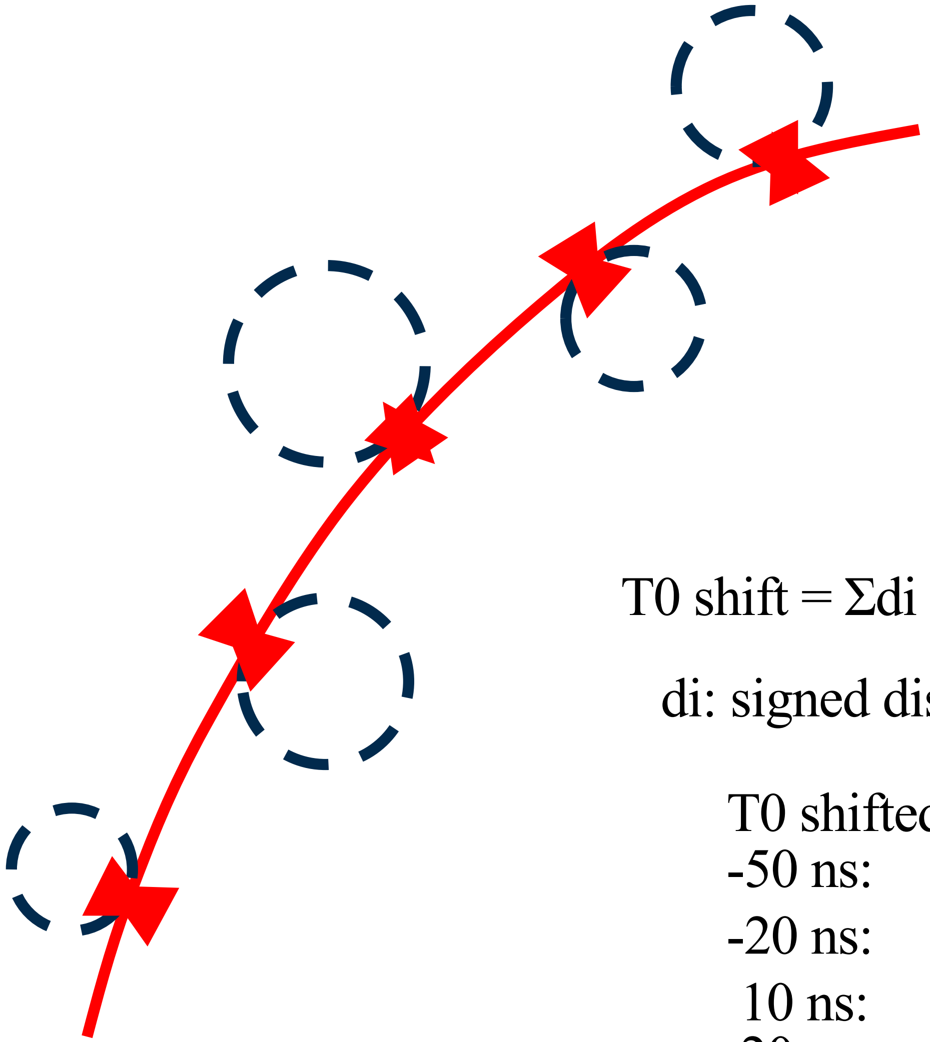
Question: To determine a line,

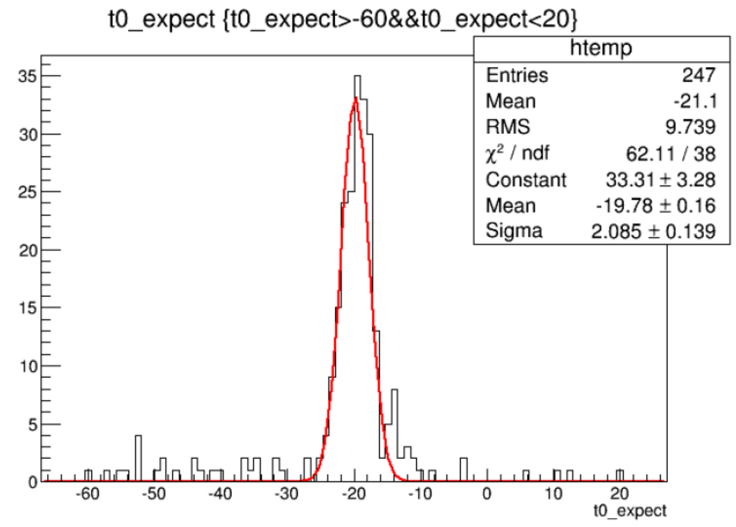
$\Phi + kz + \Phi_0 = 0$

Method: Minimize $E^2 = \sum(\Phi_i + kz_i + \Phi_0)^2 (1/d_i)^2$

$$\begin{pmatrix} S_{zz} & S_z \\ S_z & 1 \end{pmatrix} \begin{pmatrix} k \\ \phi_0 \end{pmatrix} = \begin{pmatrix} -S_{\phi z} \\ -S_{\phi} \end{pmatrix}$$

Extract T0 from Tracking





T0 shift = Σdi /N / const (N: number of hits

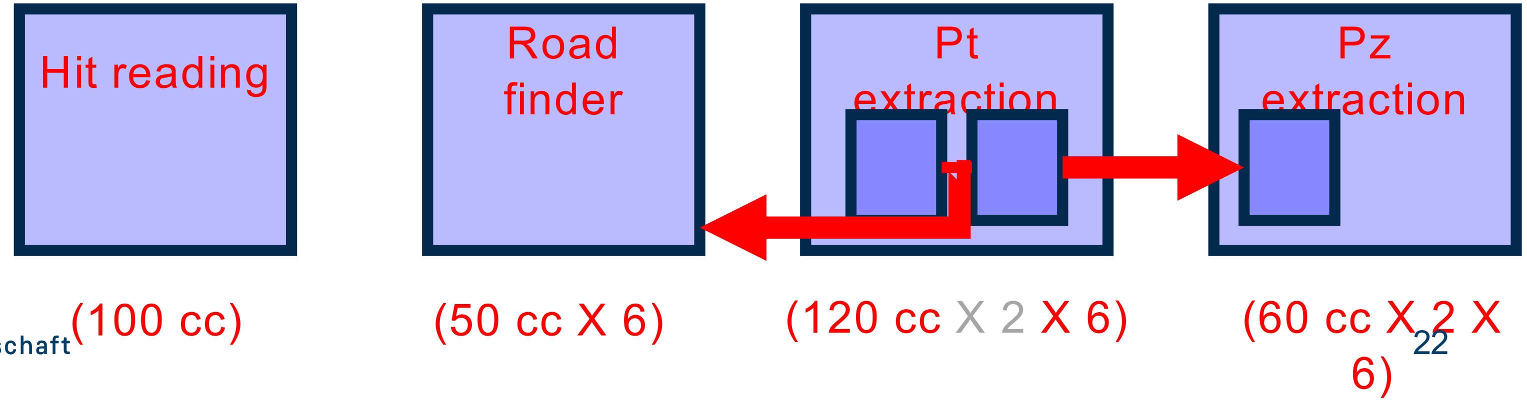
di: signed distance of circle to track)

T0 shifted by:	Extracted T0:
-50 ns:	-47.0±4.0 ns
-20 ns:	-19.8±2.1 ns
10 ns:	9.4±2.5 ns
20 ns:	19.0±2.3 ns

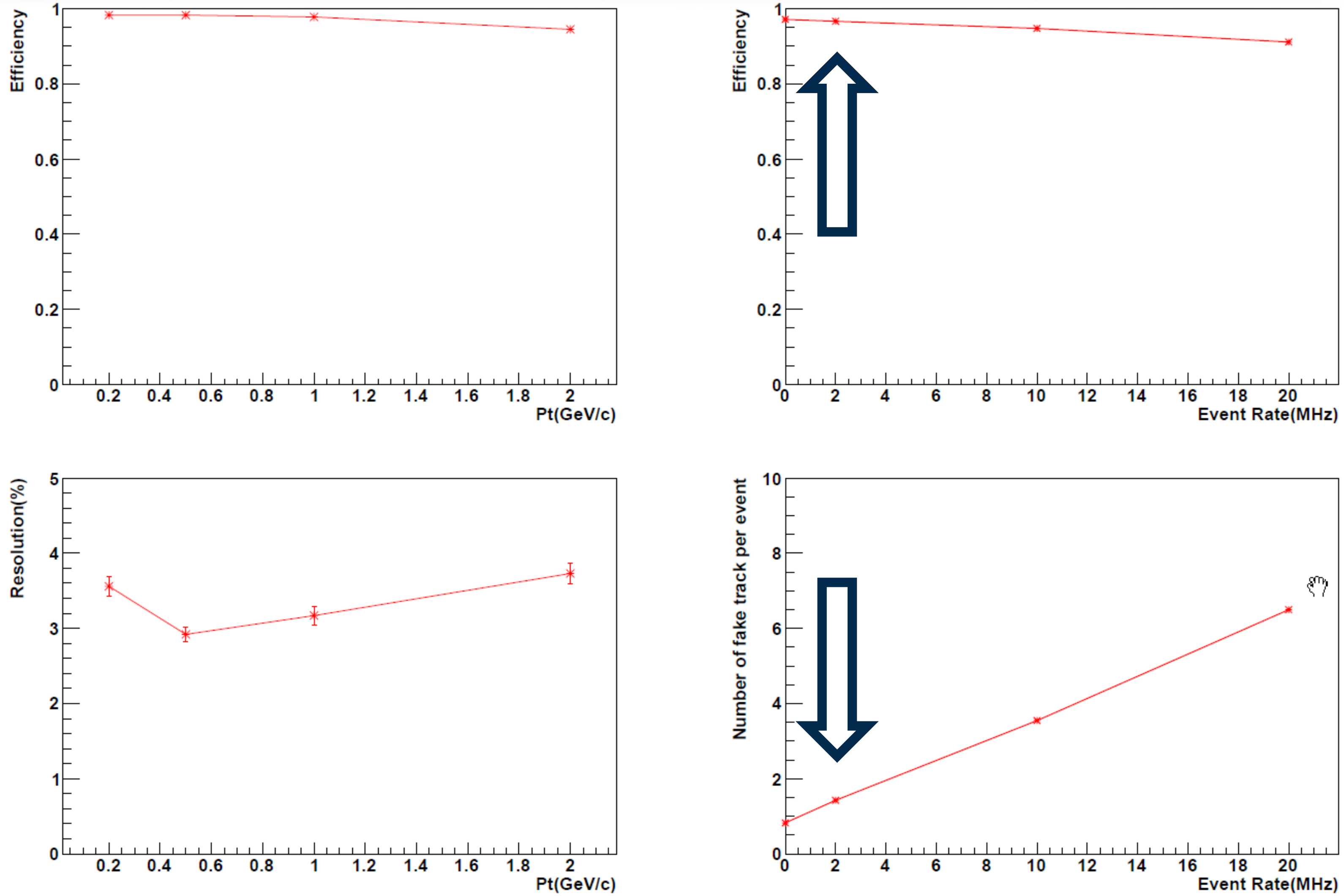
Performance at FPGA

Device Utilization Summary			
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice Flip Flops	25,022	50,560	49%
DCM autocalibration logic	14	25,022	1%
Number of 4 input LUTs	33,120	50,560	65%
DCM autocalibration logic	8	33,120	1%
Number of occupied Slices	21,563	25,280	85%
Number of FIFO16/RAMB16s	148	232	63%
Number used as RAMB16s	148		
Number of DSP48s	124	128	96%

For one event with 100 hits (6 tracks): 7 µs

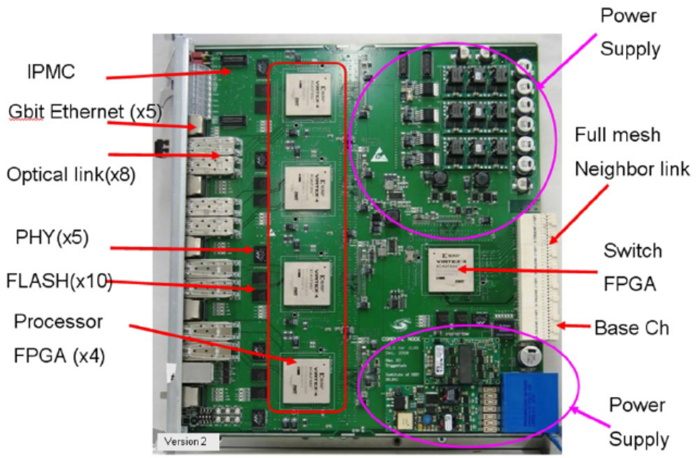


Performance test



Summary and Outlook

- σ_pt : ~ 3.2%   σ_pz : ~ 4.2% .
- 7 μs/event (6 tracks)
- Tracking strategy in case of W/O T0.



Sitcp: Ready