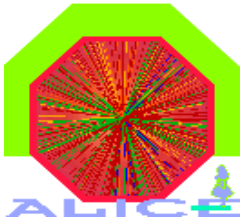


Fast track reconstruction for the TPC

Matthias Richter

Department of Physics and Technology, University of Bergen, Norway

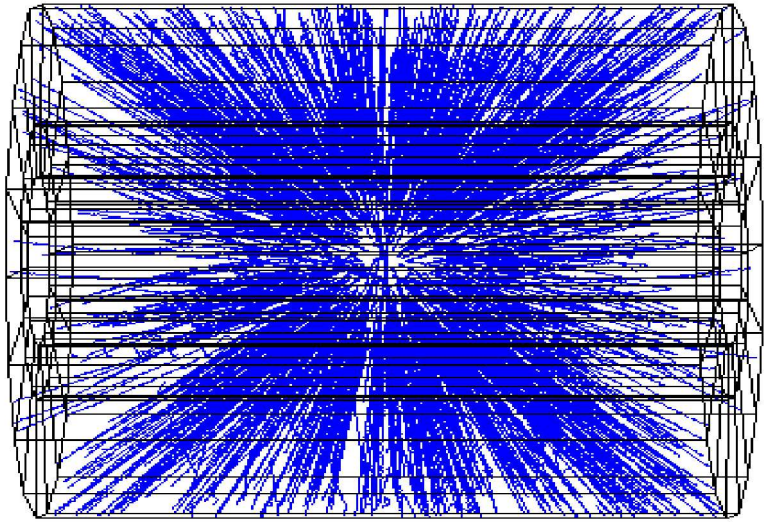
ALICE HLT/ Offline workshop, CERN Dec. 6th - 8th

Track reconstruction in the TPC

TPC occupancy:

estimation:

$dN_{ch}/d\eta$=8000 : **20000** tracks in the TPC



two approaches:

Cluster finding

Reconstruct space points from 2D clusters

Sequential tracking

Iterative tracking

Track reconstruction

Connect space points into tracks and fit them to a model (helix)

- Sequential tracking
  - Cluster finding (weighted mean)
  - Track follower
- Iterative tracking
  - Hough transform on Raw ADC-Data gives track candidates
  - Cluster fitting with respect to track parameters

Sequential tracking – the algorithms

Input: ADC-sequences above threshold

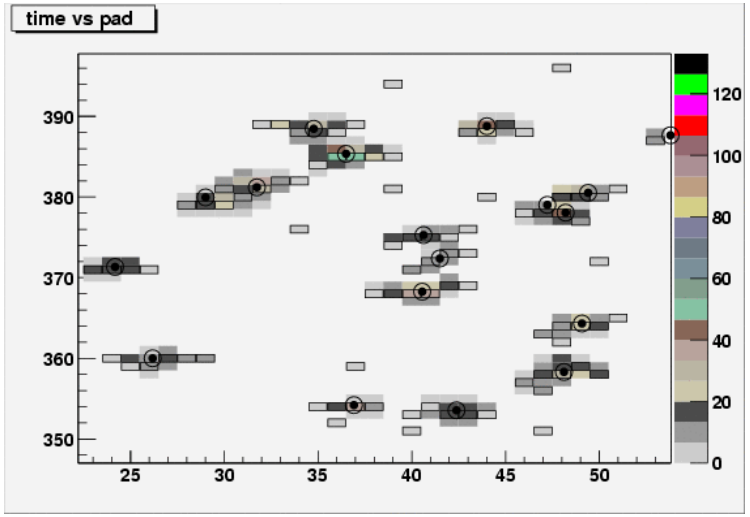
Cluster Finder

- Simple sequence matching between neighboring pads

- 2 lists in memory principle: current and previous pad(s)

- Centroids calculated as weighted mean of ADC-values

Simple deconvolution scheme:
Split clusters at local minima



Input: 3D space points

Track Finder

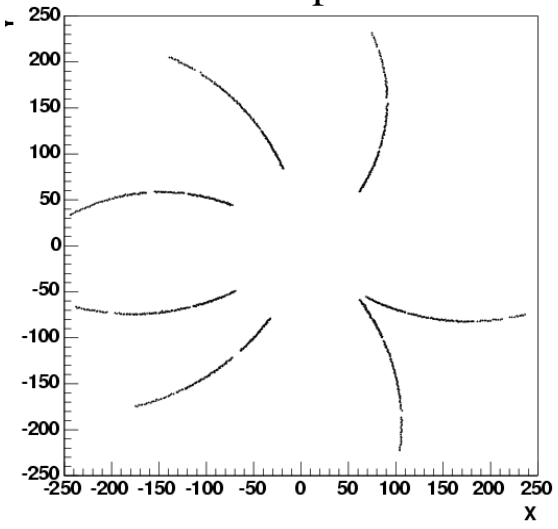
- Conformal mapping

$$x' = \frac{x - x_v}{r^2} \qquad y' = -\frac{y - y_v}{r^2}$$

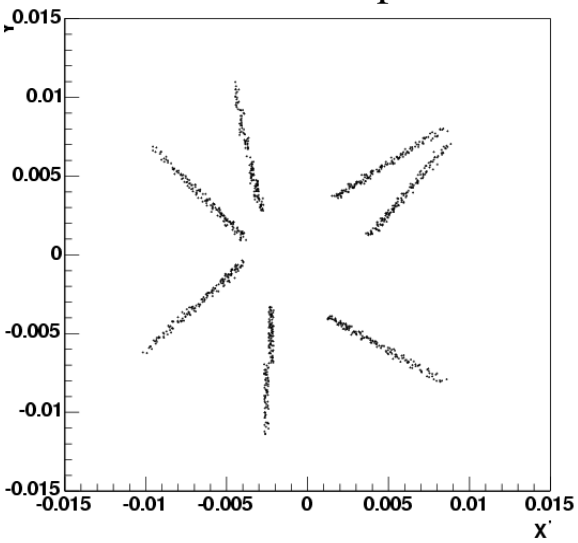
$$r^2 = (x-x_v)^2 + (y-y_v)^2$$

- "Follow-your-nose"
Build tracks from outer to inner TPC-radius

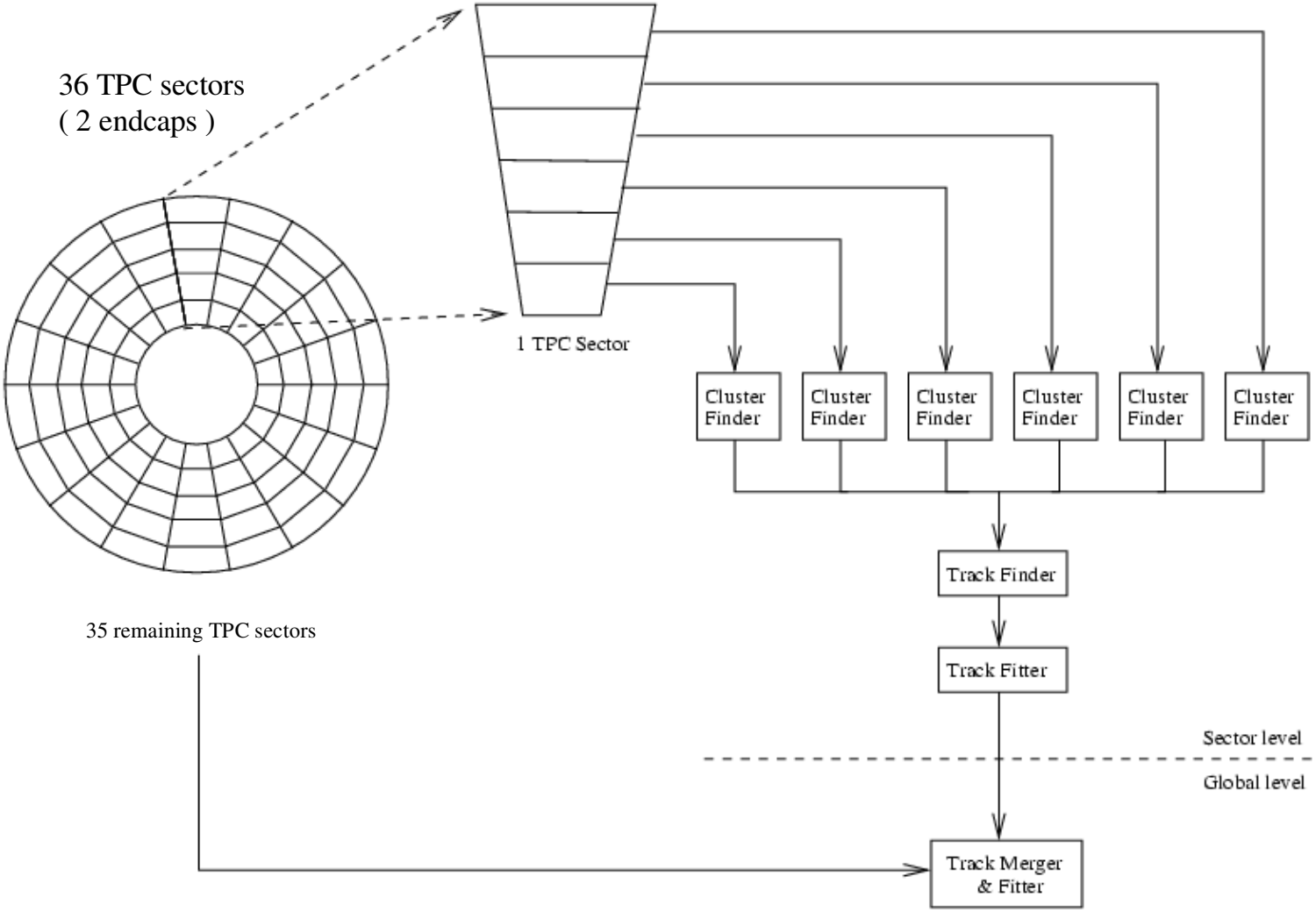
Real space



Conformal space



Sequential tracking – dataflow



Optical links

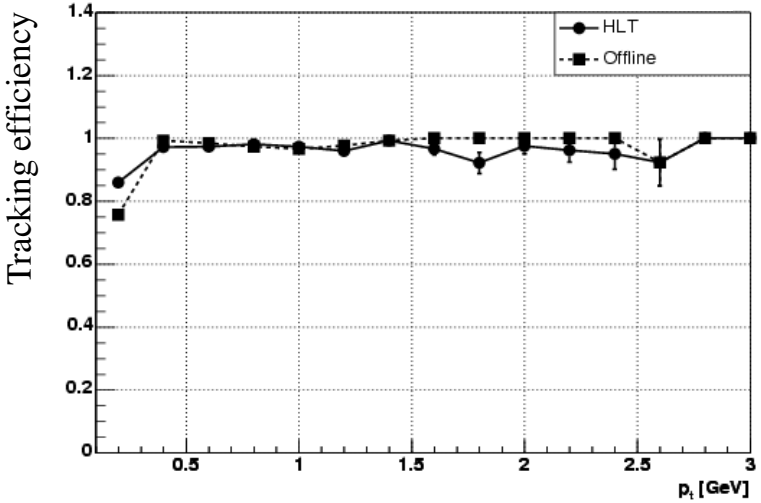
Front-End Processors:
HLT-RORC
(Readout Receiver Card)

Sector processors

Global/event processors

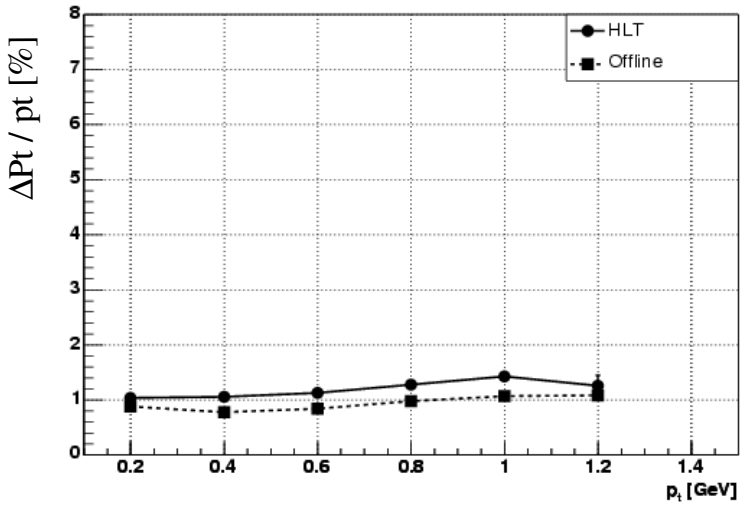
Sequential tracking – performance

Tracking efficiencies vs pt

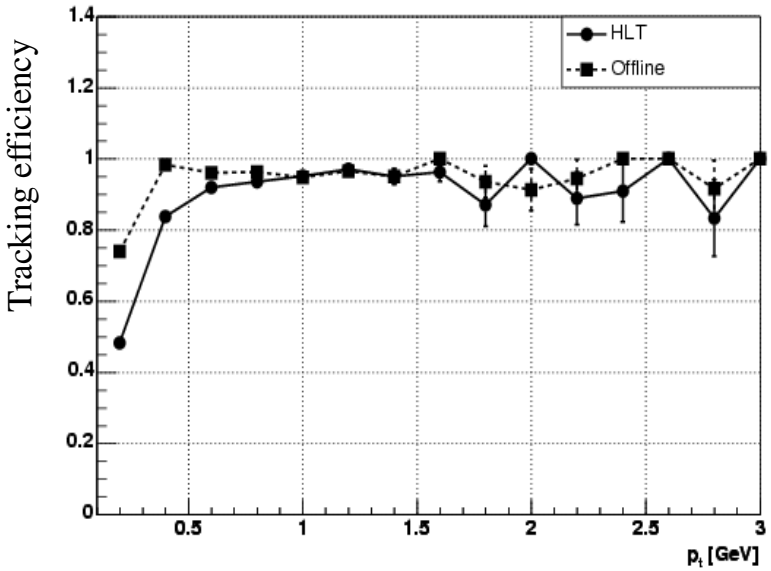


$dN_{ch}/d\eta=1000$

Transverse momentum resolutions

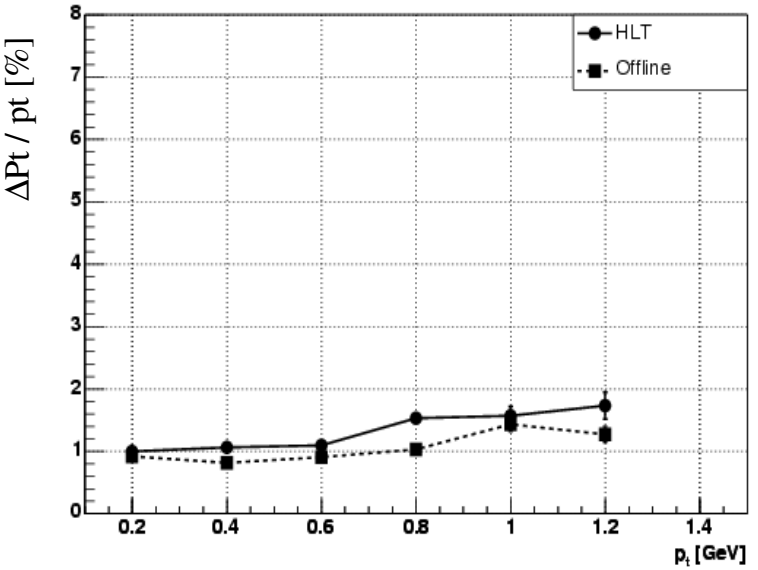


$$\text{Efficiency} = \frac{\text{Found “good” tracks}}{\text{Generated “good” tracks}}$$



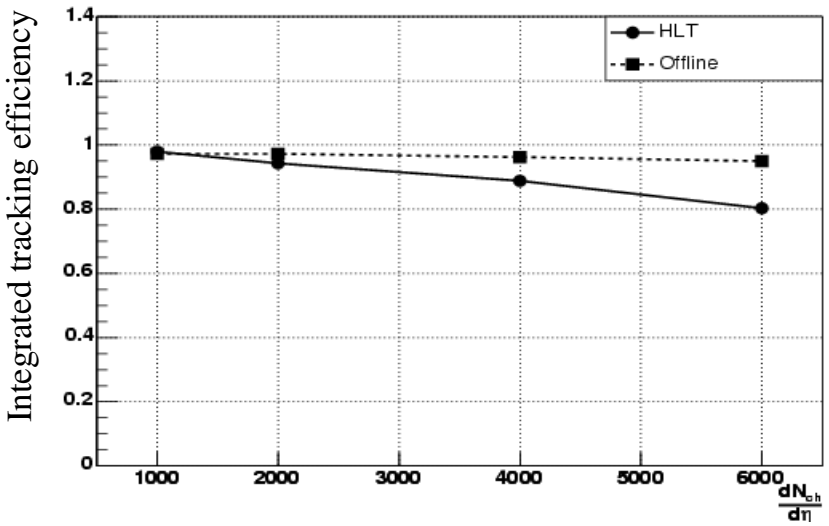
$dN_{ch}/d\eta=4000$

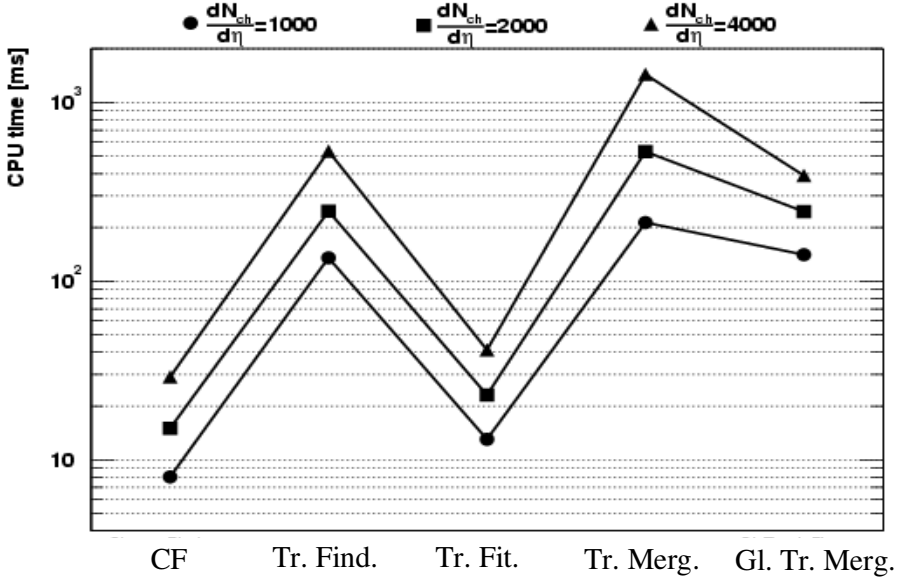
$$\Delta Pt/Pt = \frac{pt(measured) - pt(particle)}{pt(particle)}$$



Sequential tracking – performance/ computing requirements

Magnetic field: 0.2T





	Pentium III, 800 MHz		**Pentium 4, 2800 MHz**	
$dN_{ch}/d\eta$	CPU-time [s]	#CPU	CPU-time [s]	#CPU
1000	7.5	1500	3.4	680
2000	14.0	2800	6.3	1260
4000	29.5	5900	13.2	2650
6000	47.3	9460	21.2	4240

#CPU = Equivalent number of required CPUs @ 200 Hz processing rate

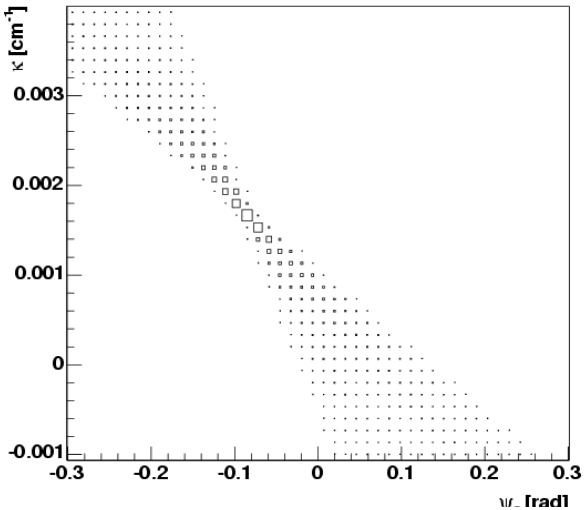
Iterative tracking – the algorithms

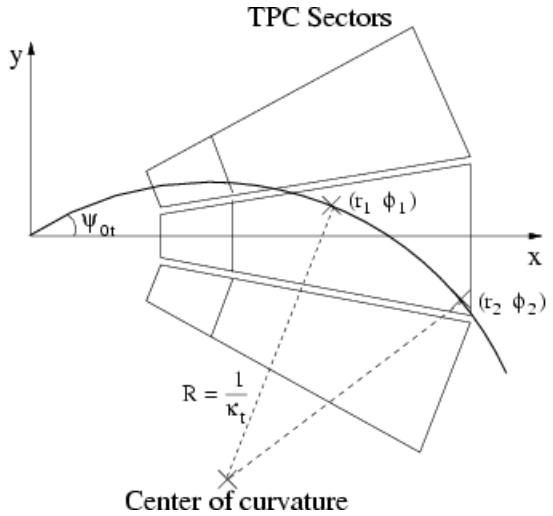
Input: ADC-sequences above threshold

Hough Transform

- Apply the circle HT directly on raw ADC-data

- Track candidates corresponds to local maxima in 2D parameter space





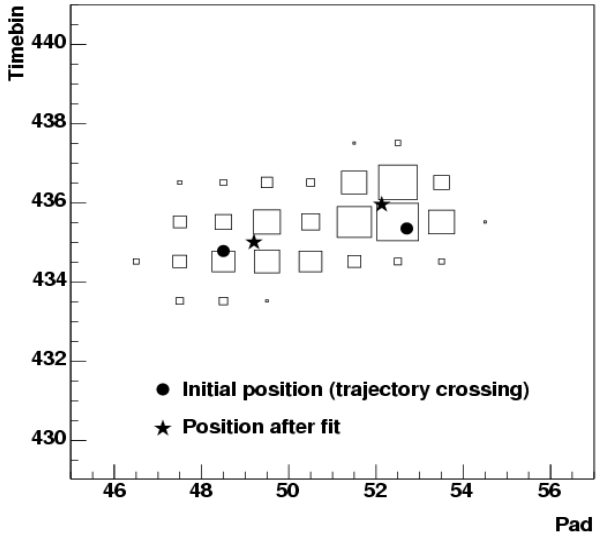
Input: Track candidates

Cluster Fitter

- Fit the clusters along the trajectories

- Track parameters provide input parameters to the fit

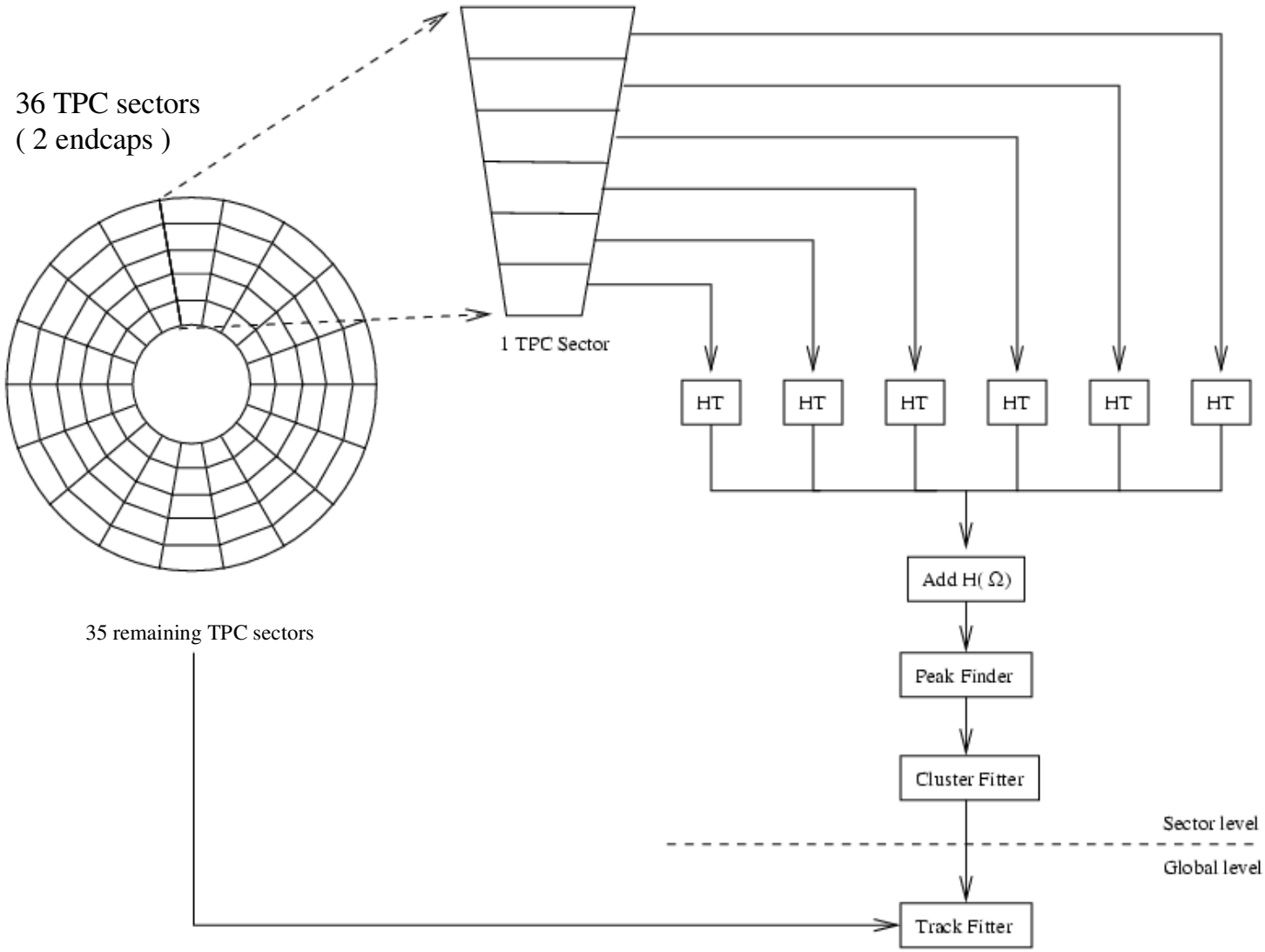
- deconvolution is done based on the knowledge of the cluster shapes



2D gauss-fit (5 parameters)

- position in pad and time
- widths in pad and time (keep fixed)
- amplitude

Iterative tracking – dataflow



Optical links

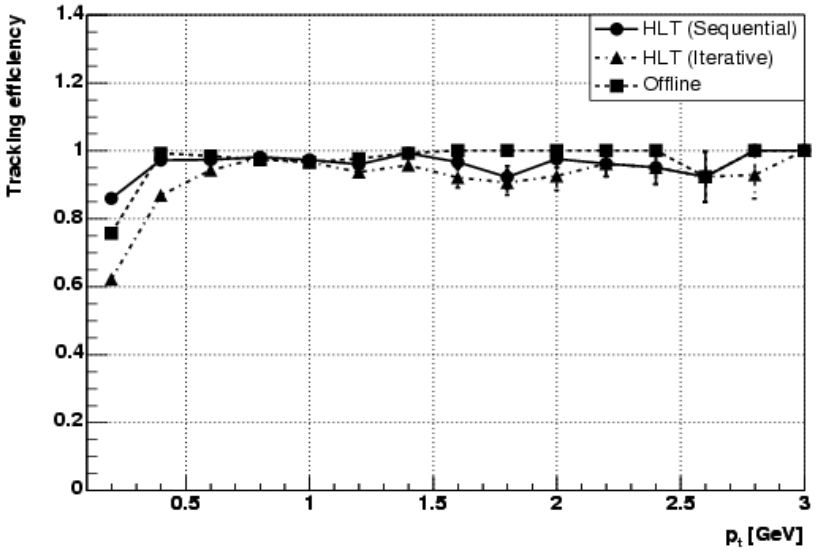
Front-End Processors:
HLT-RORC
(Readout Receiver Card)

Sector processors

Global/event processors

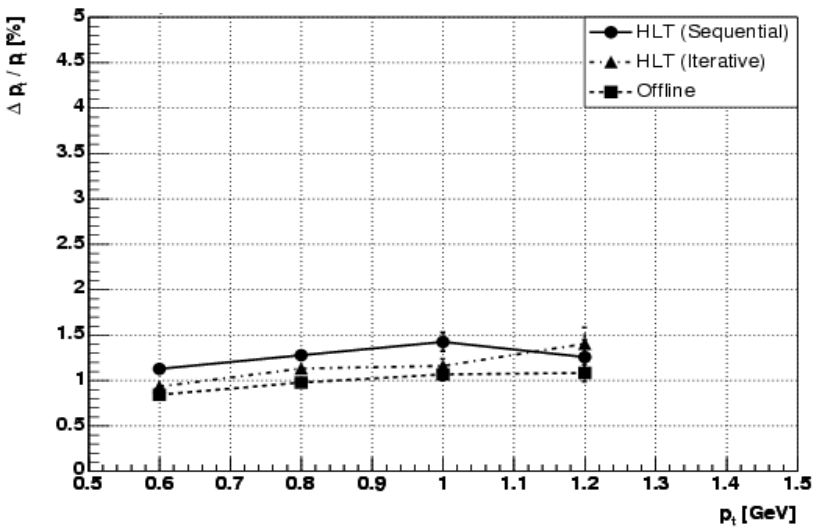
Iterative tracking - performance

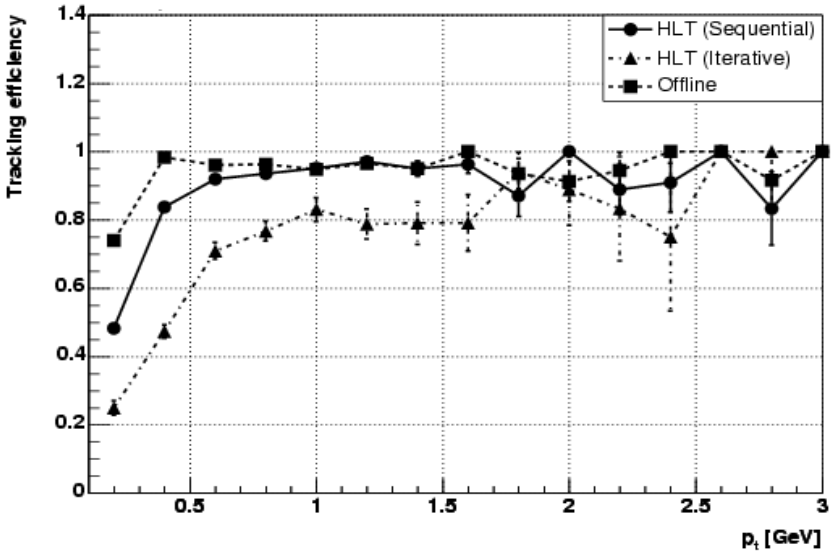
Tracking efficiencies



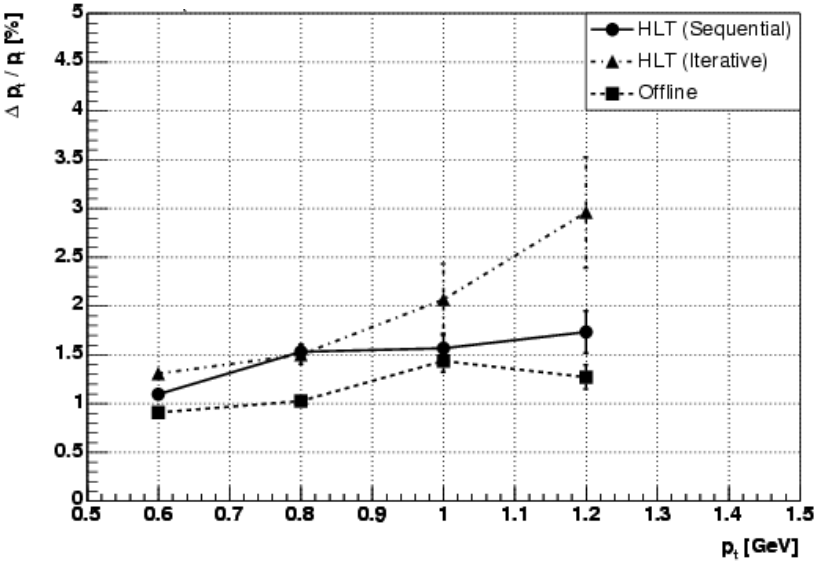
$dN_{ch}/d\eta=1000$

Transverse momentum resolution





$dN_{ch}/d\eta=4000$



Efficiency loss due to high number of "false" peaks in HT

Iterative tracking – computing requirements

Hough Transform: ($dN_{ch}/d\eta$=4000)

3.5 s per sub-sector * 6 * 36

= 750 s for complete TPC event

→ 150 000 CPUs @ 200 Hz processing rate (PIII 800 MHz)

Heavy IO-bound algorithm, as extensive and repetitive access to memory is needed (filling histograms).

- Algorithm is however local by nature, high degree of parallelization.
- ongoing development of fast algorithm, C. Cheshkov

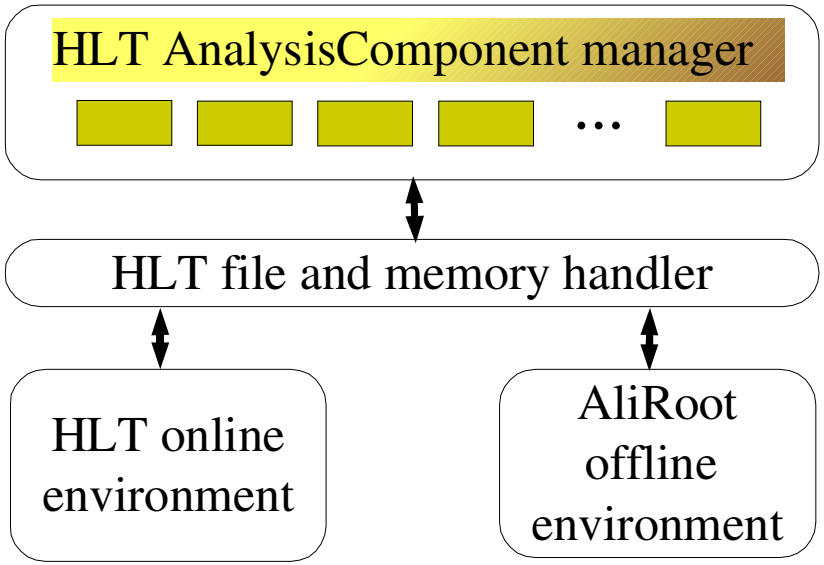
↓

Suitable for implementation in FPGA co-processor on the HLT-RORC.

Status of analysis components for TPC

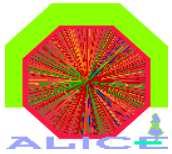
- Fully implemented and tested
- Analysis components developed within AliROOT
- Data internally organized in simple C structures to minimize overhead
- There is already some kind of abstract interface, the file and memory handler

  -> connects analysis components to either online or offline framework



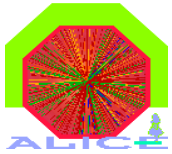
However:

- realization of analysis chain in AliROOT still “hand-knitted”
- “user “ of online framework needs to know a lot about pub/sub framework to implement components



Issues and open questions

- we have to move from prototype implementations to a “production version”
- need uniform policy how to build an HLT analysis chain applicable to all sub detectors
- analysis component concept
- better separation of transport and analysis components in the online framework



Acknowledgment

- Talk based on work of Anders Vestbø
- Algorithms developed and implemented by A. Vestbø, C. Loizides, U. Frankenfeld
- further members of the HLT group: T. Alt, H. Helstrup, V. Lindenstruth, G. Øvrebekk, D. Röhrich, B. Skaali, T. Steinbeck, R. Stock, H. Tilsner, T. Vik