Tracker Development

I: Introduction
II: GPU Tracker Performance
III: CPU / GPU Tracker Comparison
IV: Global Tracking

David Rohr – Frankfurt Institute for Advanced Studies

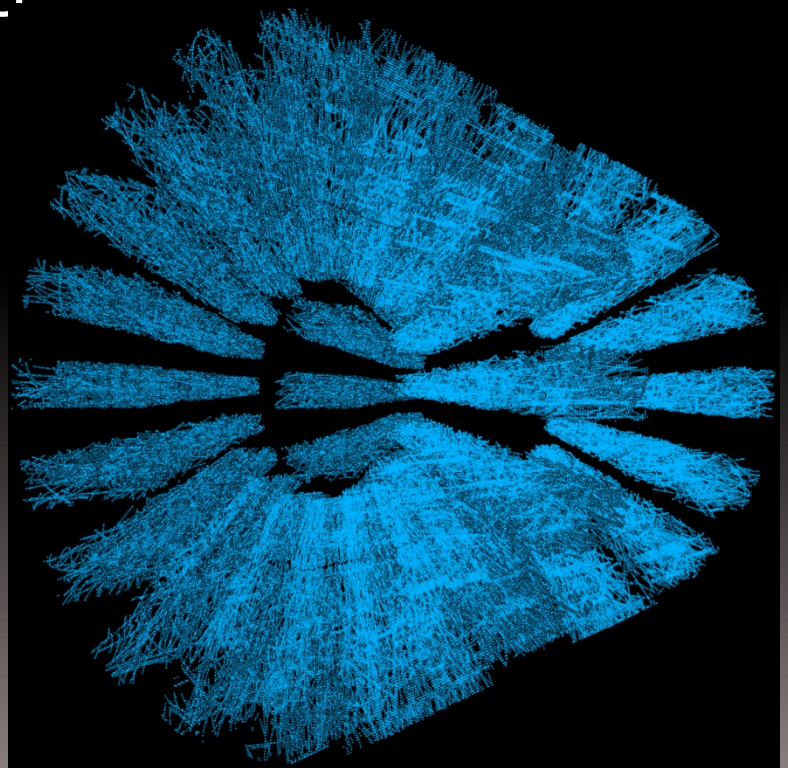
27.2.2012

Introduction

- Original cellular automaton tracker developed by Sergey Gorbunov.
- Alogorithm was designed for parallelism from the beginning.
- A GPU enabled version for the GTX285 was developed in 2010.
- Flaws of the old version, improvements, and a new version for the Fermi GPU will be presented.

Introduction

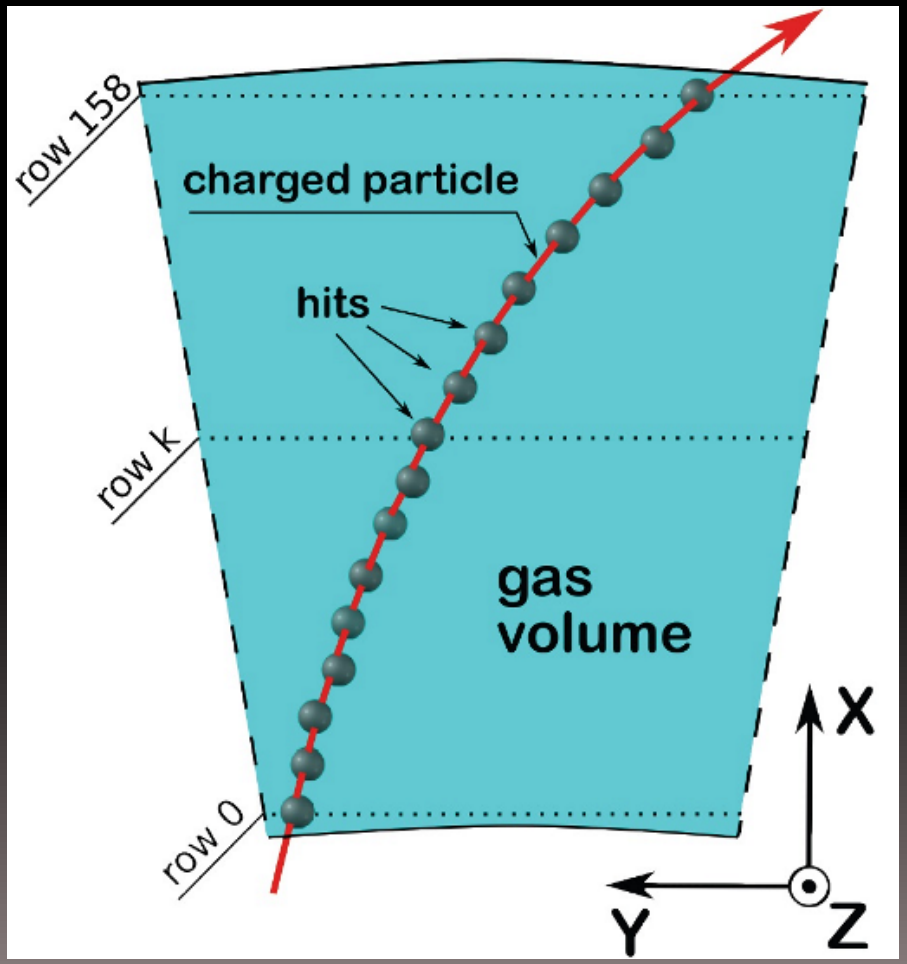
- ALICE HLT tracker divides the TPC in slices and processes the slices individually.
- Track Segments from all slices are merged later.



Introduction

One ALICE TPC Sector

(Traking is performed row by row)



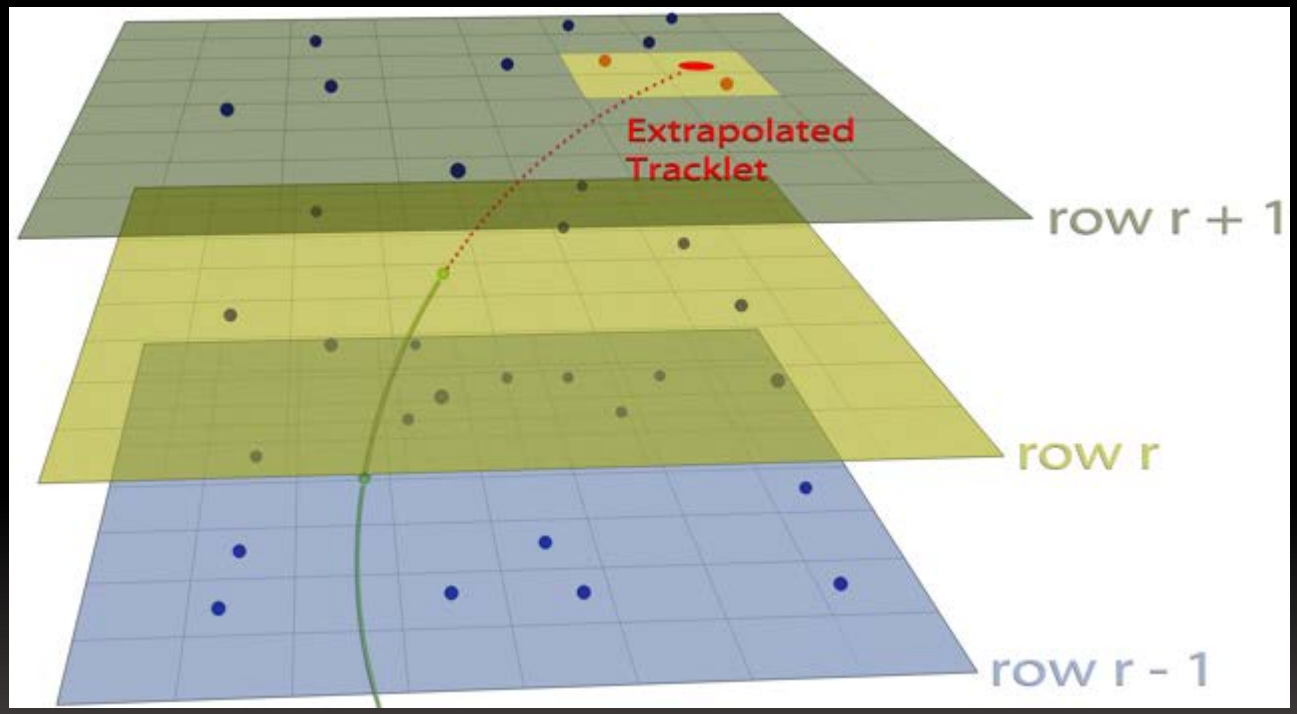
Introduction

Tracking Algorithm

Category of Task	Name of Task	Description on Task
	(Initialization)	
Combinatorial Part (Cellular Automation)	I: Neighbors Finding	Construct Seeds (Track Candidates)
	II: Evolution	
Kalman Filter Part	III: Tracklet Construction	Fit Seed Extrapolate Tracklet Find New Clusters
	IV: Tracklet Selection	Select good Tracklets, Assign clusters to tracks
	(Tracklet Output)	

Introduction

Step III: Tracklet Construction

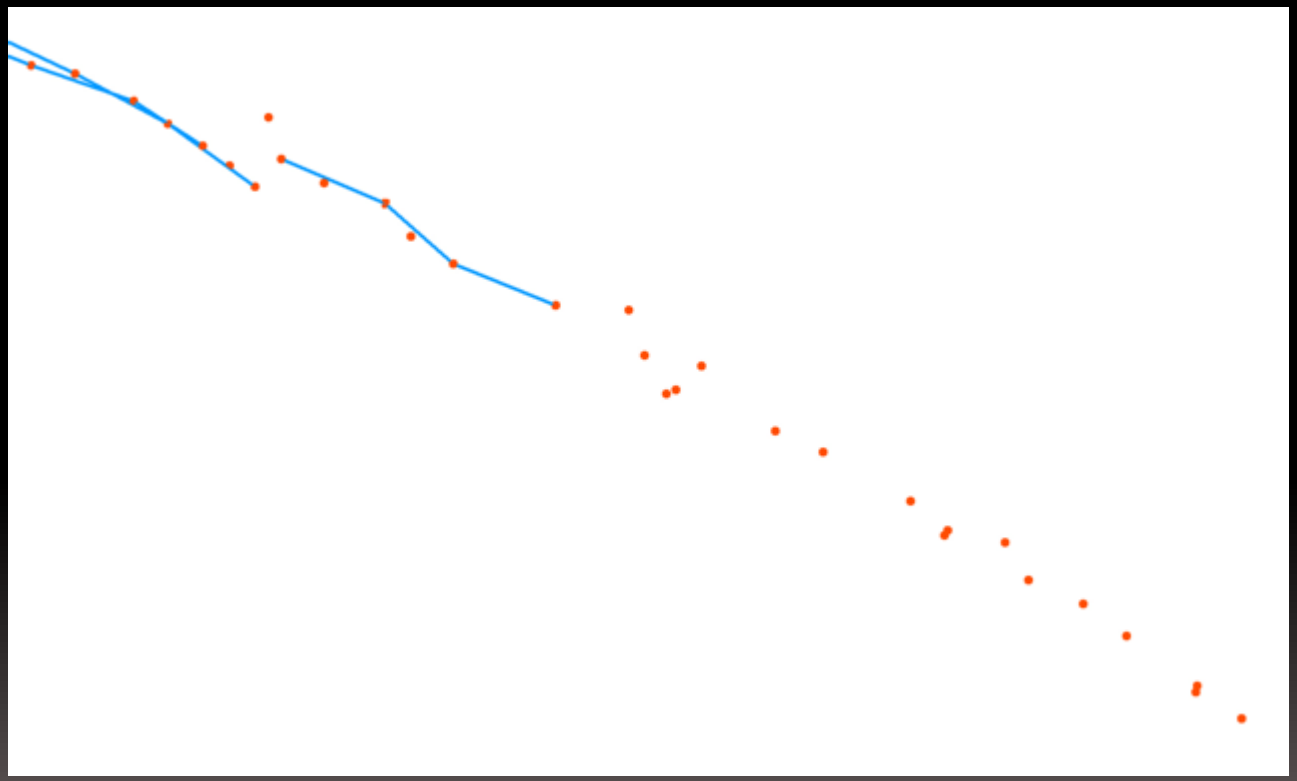


Green: Seed Red: Extrapolation

Clusters close to the extraplation point are searched

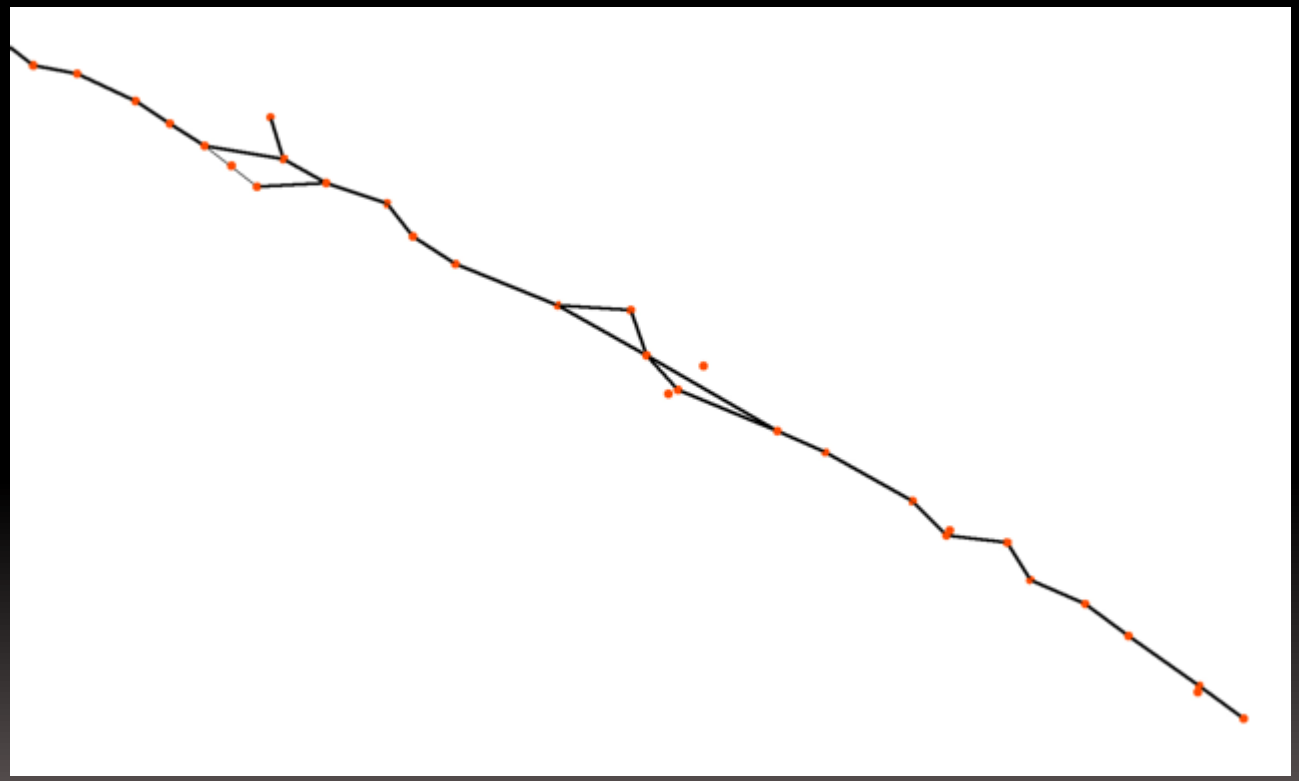
Introduction

Evolution Step on Real Data



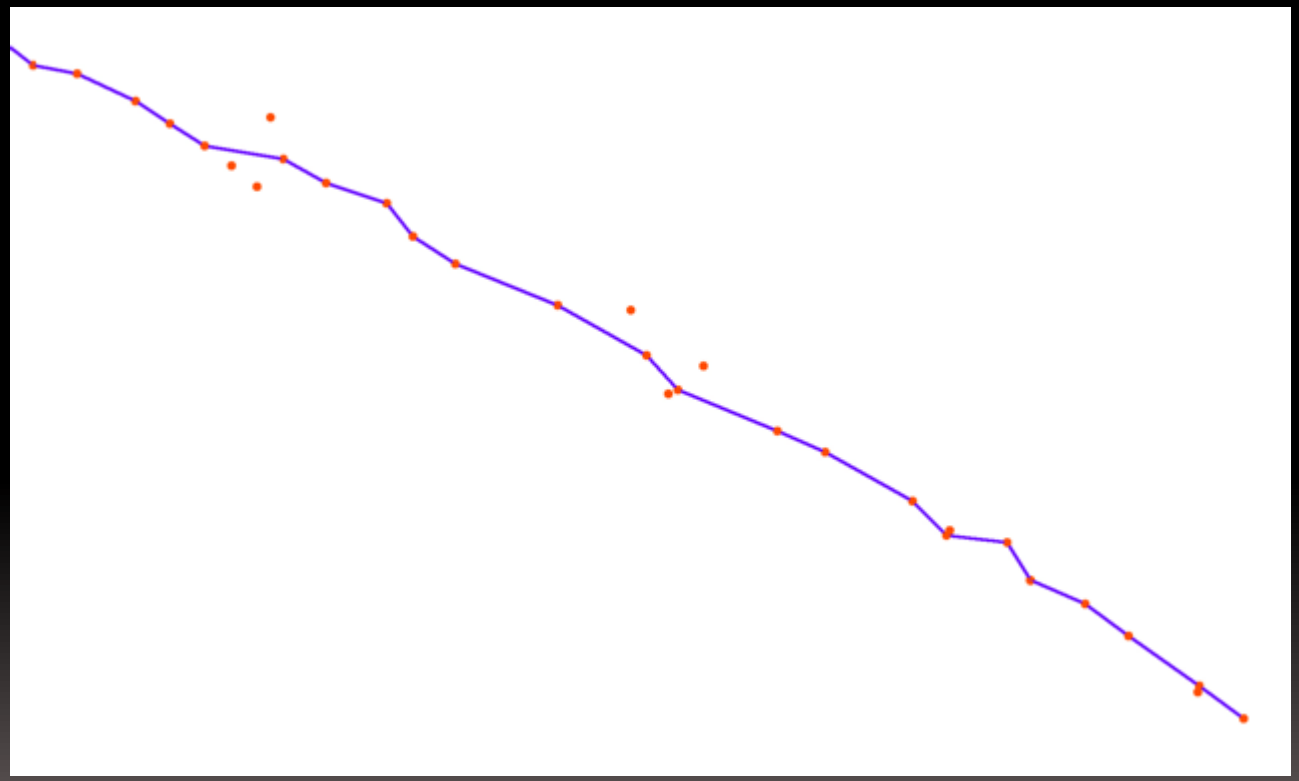
Introduction

Tracklet Construction on Real Data



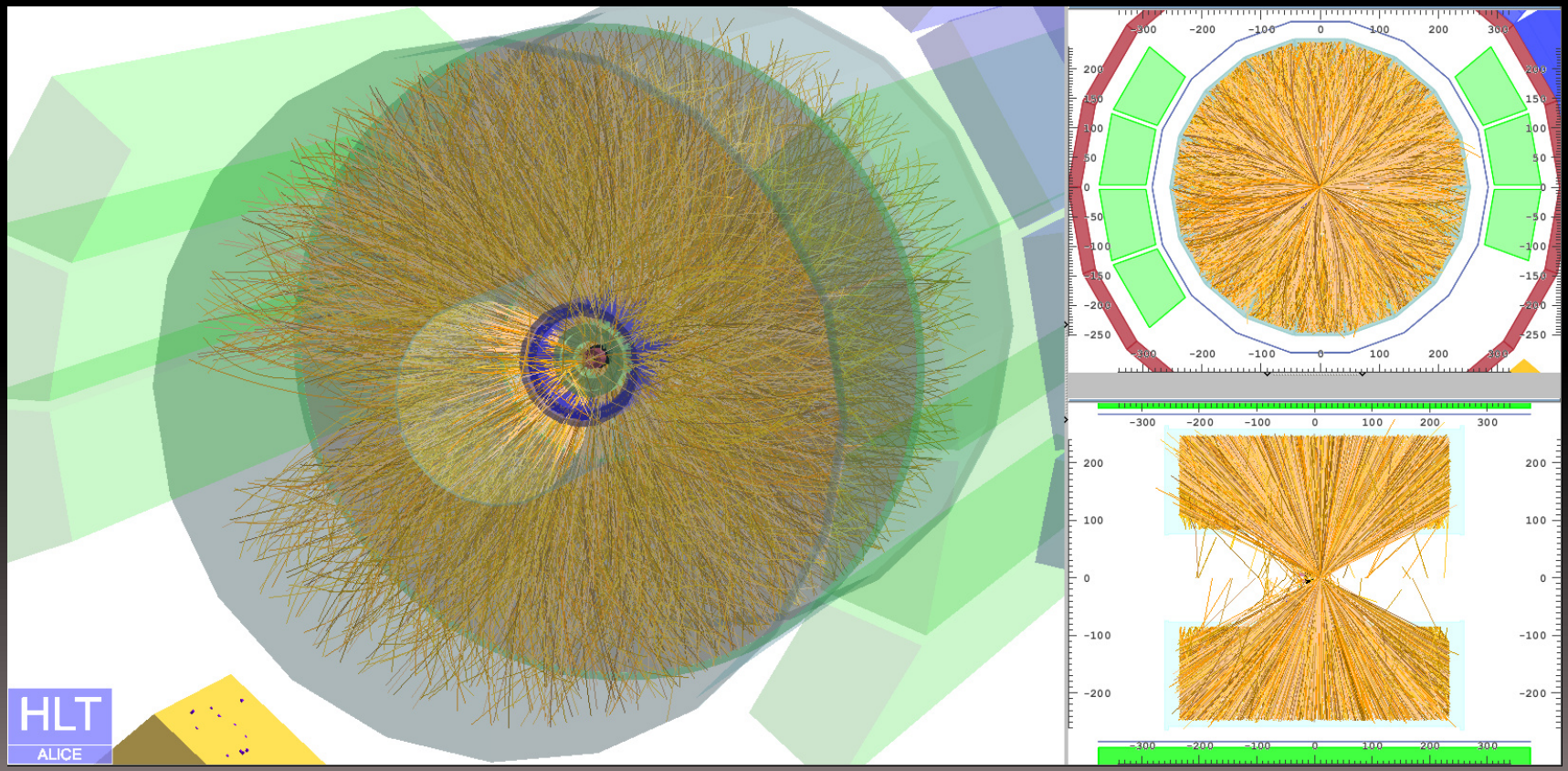
Introduction

Tracklet Selection on Real Data



Introduction

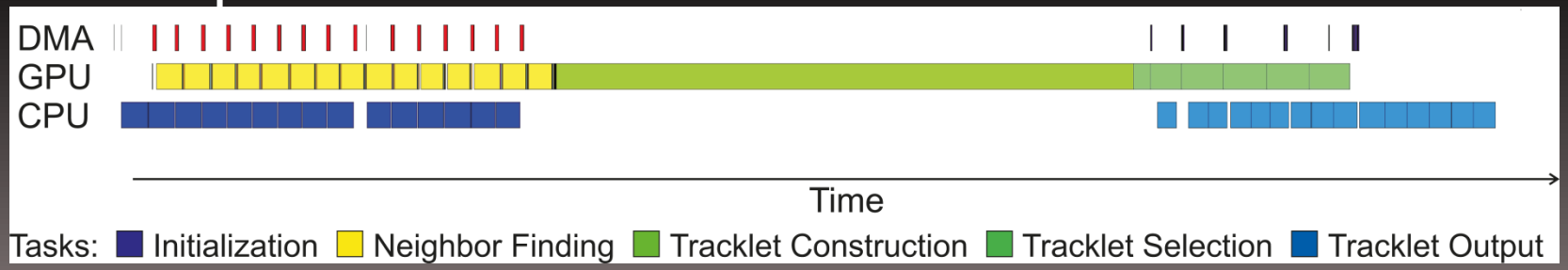
Screenshot of Event-Display during first physics-fill with active GPU Tracker



CPU/GPU PERFORMANCE

GPU Tracker Performance

- For good performance the GPU tracker pipelines the slices such that initialization on CPU, GPU tracking, and DMA transfer can overlap.
- Pipeline on old hardware works well, initialization on CPU and first GPU step require similar time.

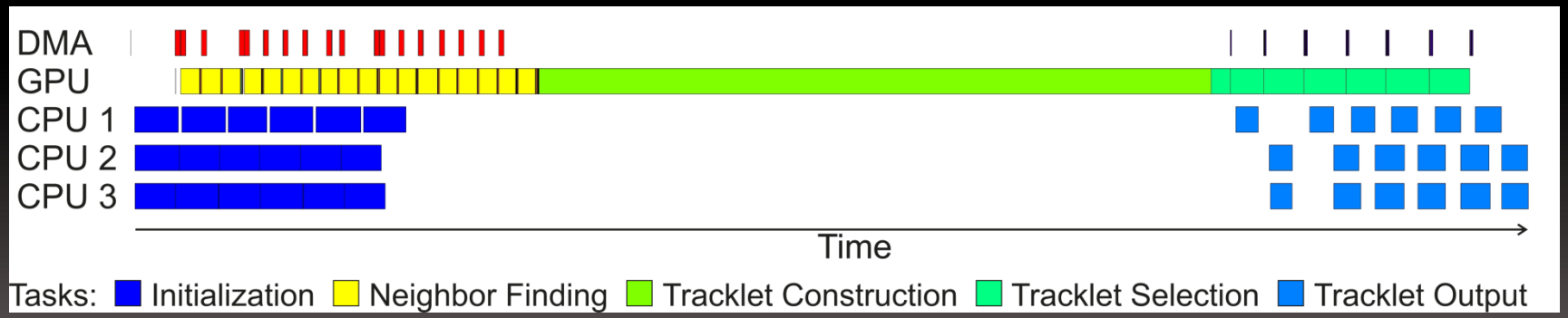


GPU Tracker Performance

- On new hardware, Fermi GPU and Magny-Cours CPU, simple pipeline does not work
  - Per-core performance of Magny-Cours is lower than for Nehalem (even though total peak is better), but the GPU tracker was single-threaded
  - New Fermi GPU accelerates GPU tracking

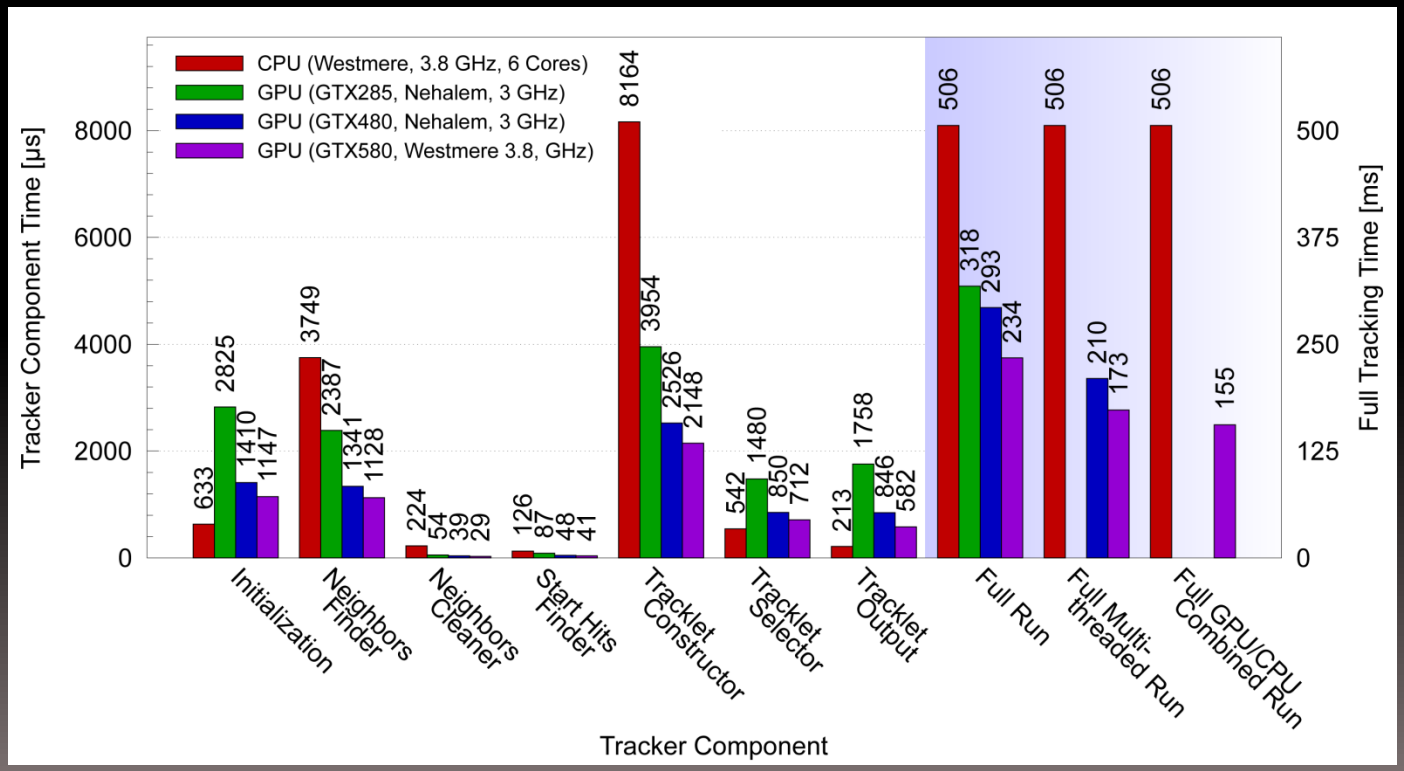
GPU Tracker Performance

- Solution:
  - Multiple GPU cores used in the pipeline.
  - The CPU threads process the slices in a round-robin fashion



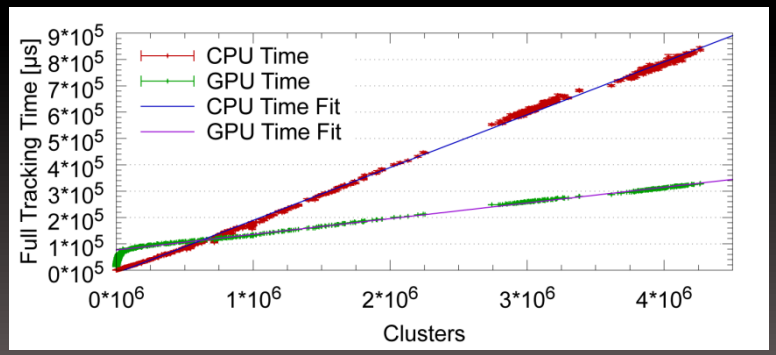
CPU / GPU Tracker Comparison

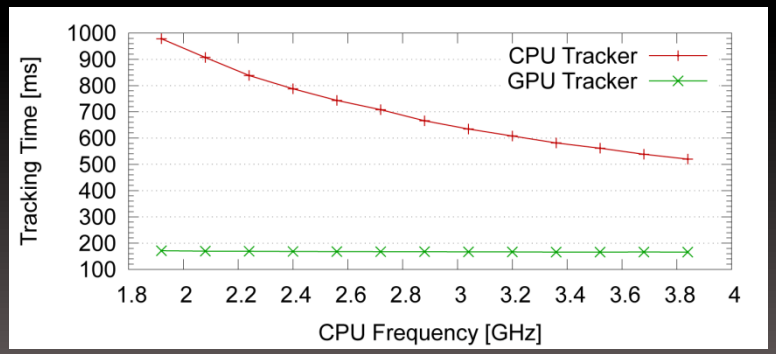
- Performance: GTX580 GPU almost three times as fast as 6-core processor.



GPU Tracker Performance

- Tracking time depends linearly on input data size.
- GPU tracking time independent of CPU performance (if initialization is fast enough).

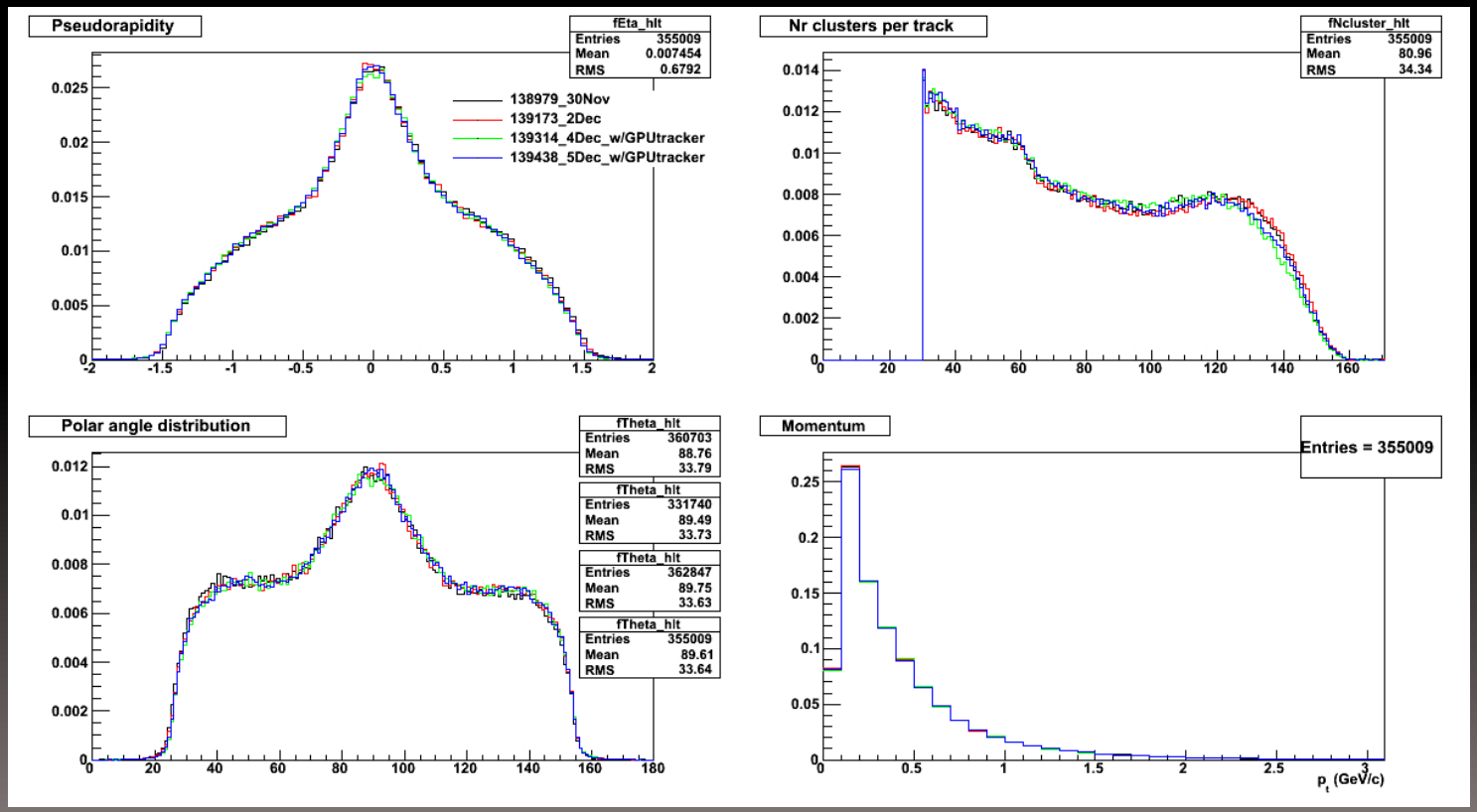




CPU / GPU TRACKER COMPARISON

CPU / GPU Tracker Comparison

- Comparison of GPU and CPU Tracker during 2010 run

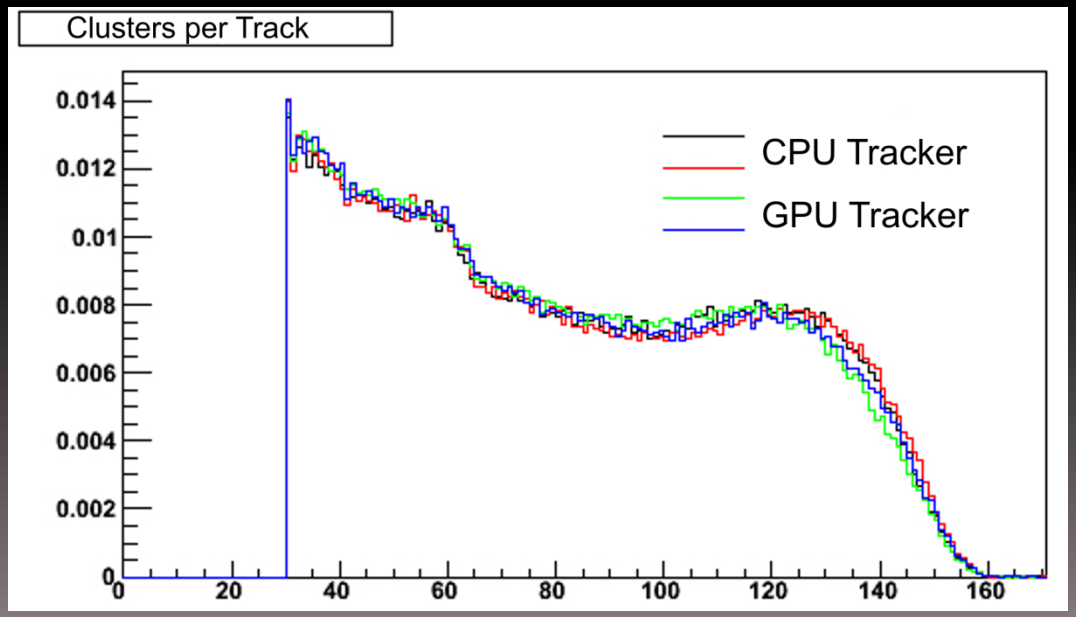


CPU / GPU Tracker Comparison

- Comparison of GPU and CPU Tracker during 2010 run
  - No significant variations in physically observables.
  - Only the number of clusters per track statistics shows a variation.

CPU / GPU Tracker Consistency

- Inconsistencies during November 2010 run
  - Cluster to track assignment
  - Track Merger
  - Non-associative floating point arithmetics



CPU / GPU Tracker Consistency

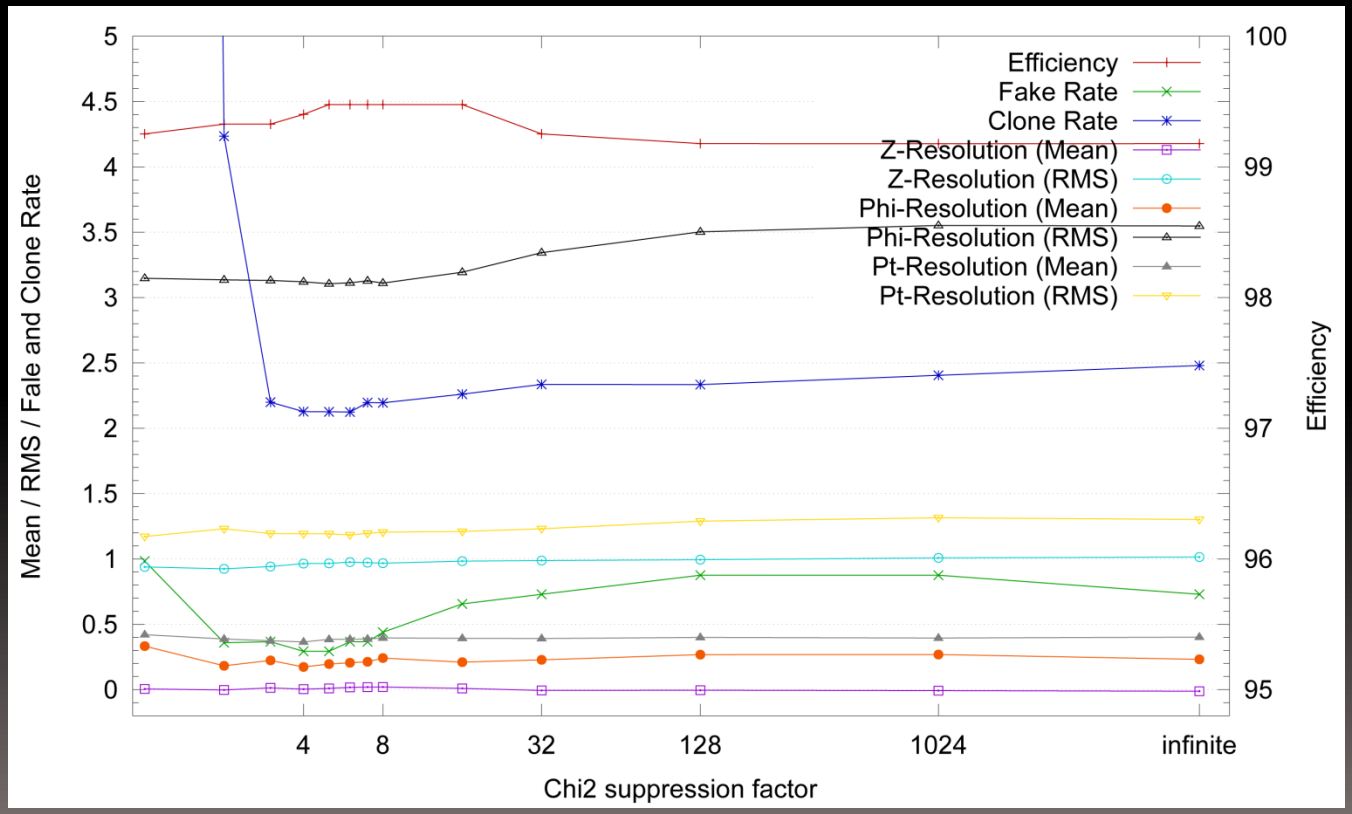
- Cluster to Track Assignment
  - Problem: Cluster to track assignment was depending on the order of the tracks.
    - Each cluster was assigned to the longest possible track. Out of two tracks of the same length, the first one was chosen.
    - Concurrent GPU tracking processes the tracks in an undefined order.
  - Solution: Both the chi^2 and the track lenth are used as criteria. It is extremely unlikely that two tracks coincide in both values.

CPU / GPU Tracker Consistency

- How to combine chi² and track length?
  - Regarding the deviation between the track and the cluster for each cluster individually leads to many clones.
  - Hence, the total deviation of the track is used.
  - Small tracks have a higher probability for having a small chi², the right weight for both parameters must be determined.
  - Therefore, a chi² suppression factor is introduced, that weigths chi² less than the tracklet length.

CPU / GPU Tracker Consistency

- Determinining best suppression factor



CPU / GPU Tracker Consistency

- Determinining best suppression factor
  - A factor of infinite equals the old method were only the track length is decisive.
  - Incorporating chi² improves efficiency and resolution.
  - At low suppression factor only the chi² is decisve and the tracking becomes unstable.
  - Currently, a factor of 6 is used.

CPU / GPU Tracker Consistency

- Track Merger
  - Problem: Result of the Track merger depended on the order of input tracks.
  - Solution: Merger input is sorted.
    - Sorting is performed during a reformatting step.

    → No additional data copy.

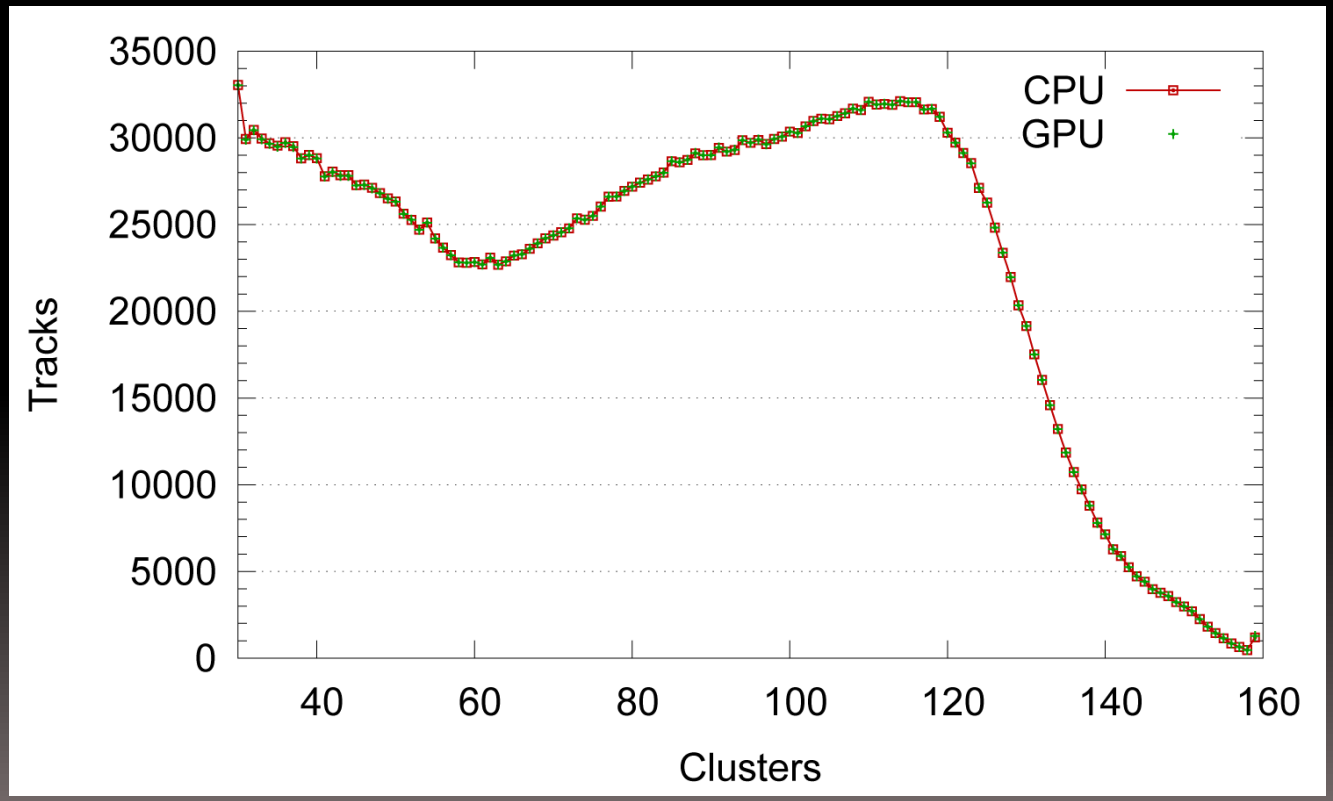
    → No performance penalty.

CPU / GPU Tracker Consistency

- Non associative floating point arithmetics
  - Problem: Different compilers perform the arithmetics in different order (also on the CPU).
  - Solution: Cannot be fixed, but…
    - Slight variations during the extrapolations do not matter as long as the clusters stay the same.
    - Inconsistent clusters: 0,00024%

CPU / GPU Tracker Consistency

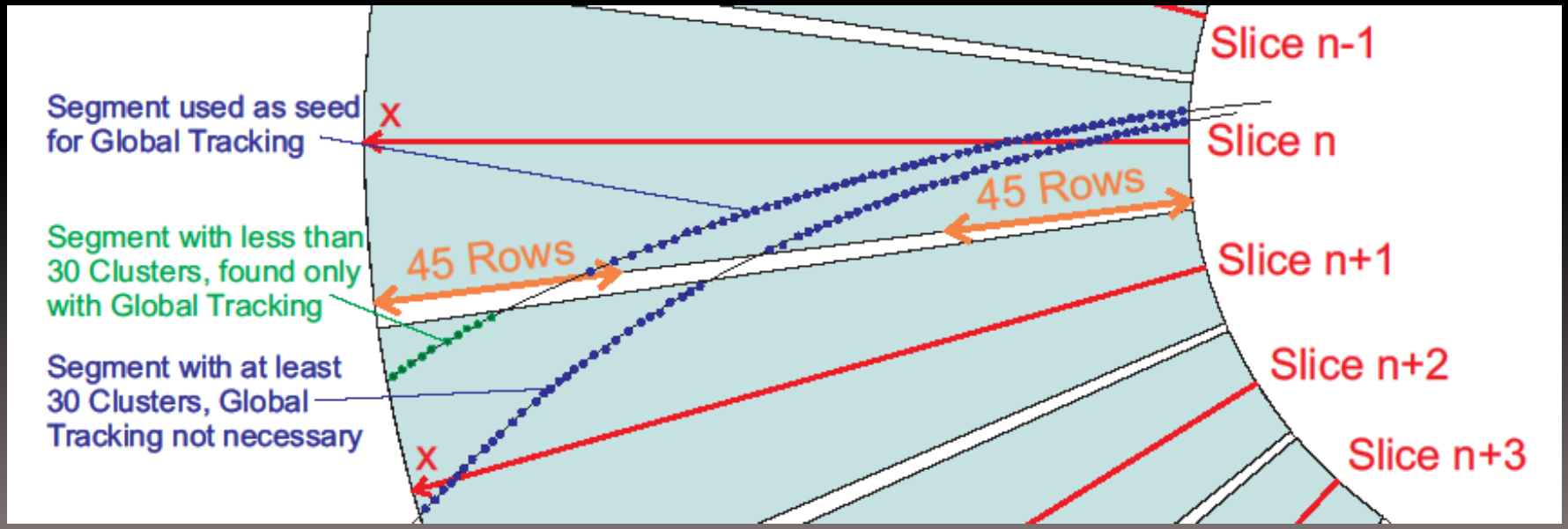
- Cluster per track statistic with improvements



GLOBAL TRACKING

Global Tracking

- Original HLT tracker did not find track segments of less than 30 clusters in a slice.
- An additional step before the merger can find these segments.

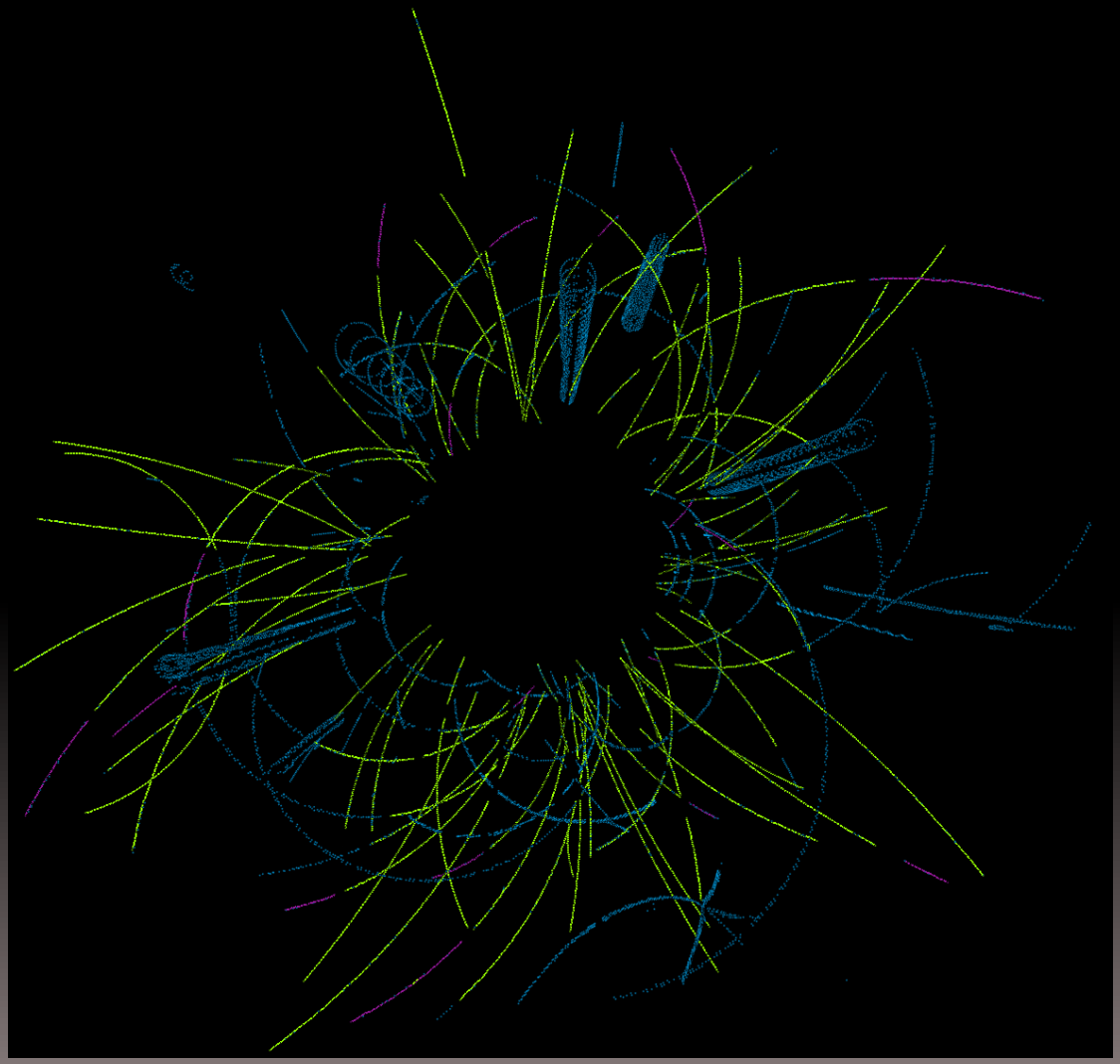


Global Tracking

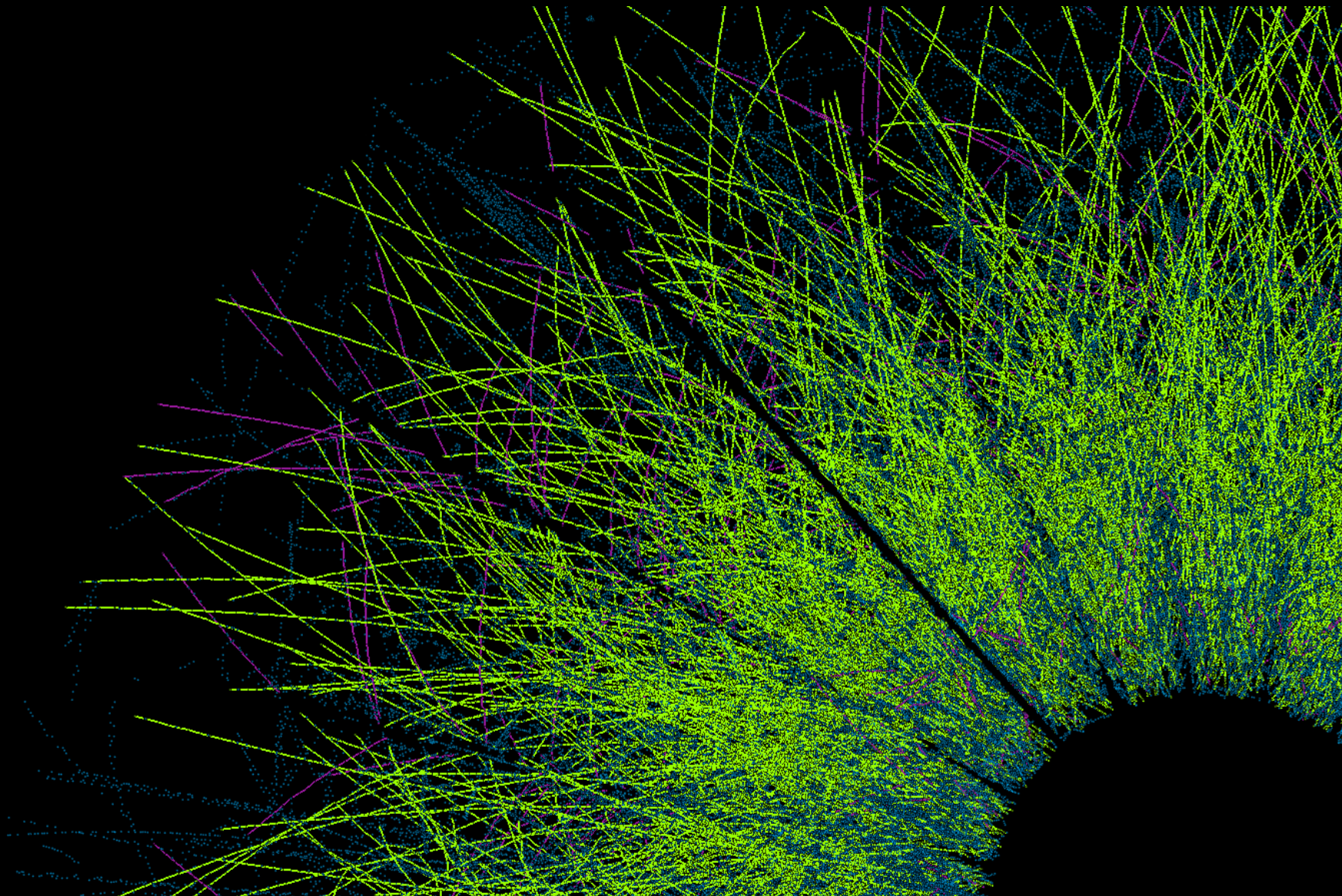
- No additional tracks found.
- Newly found track segments automatically merged with the track used as seed.
- Efficiency / clone- / fake-rate unchanged.

Global Tracking

- PP Event
- Original segments green
- Additional segments violet

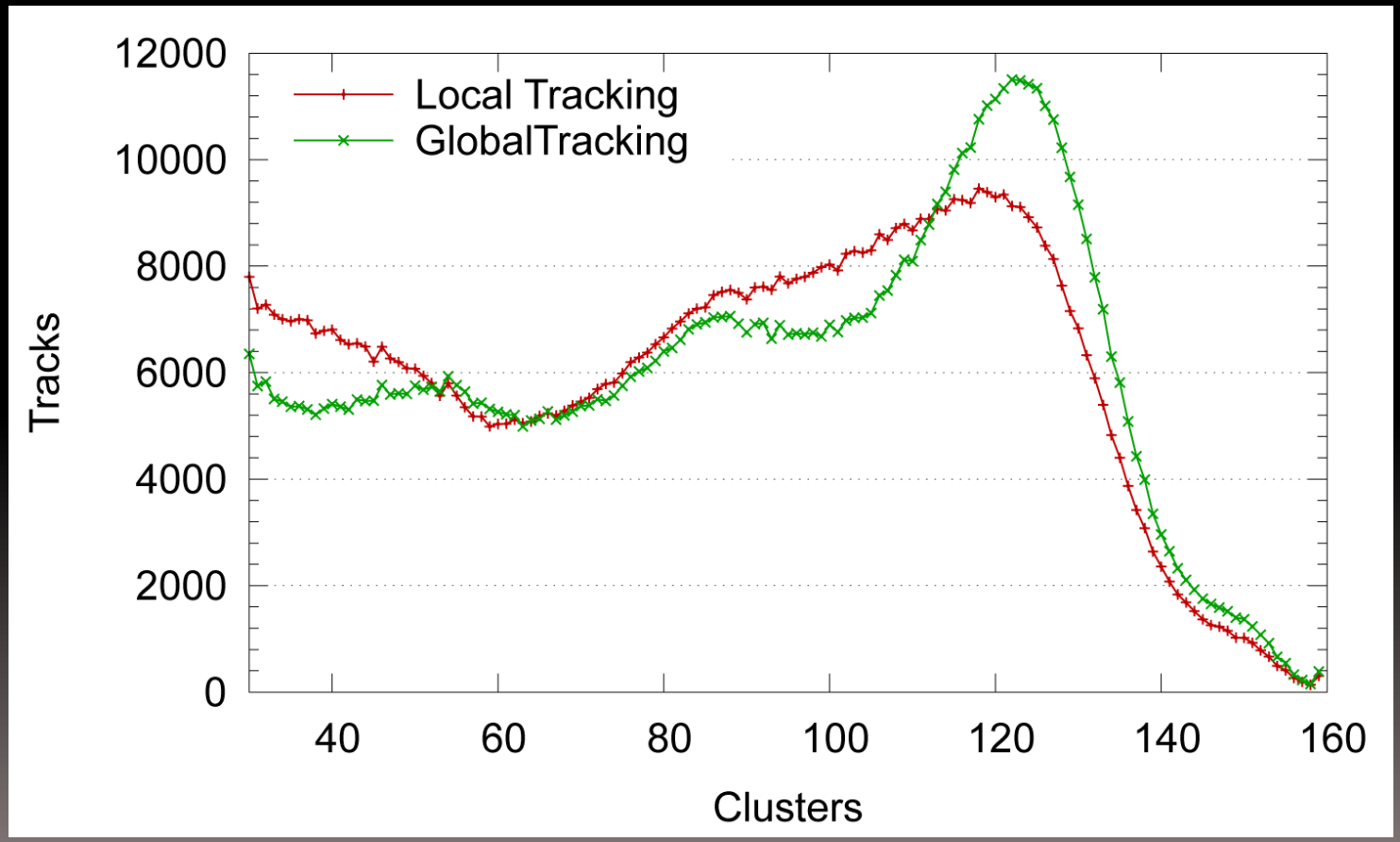


Global Tracking



Global Tracking

- Cluster per track statistics comparison:



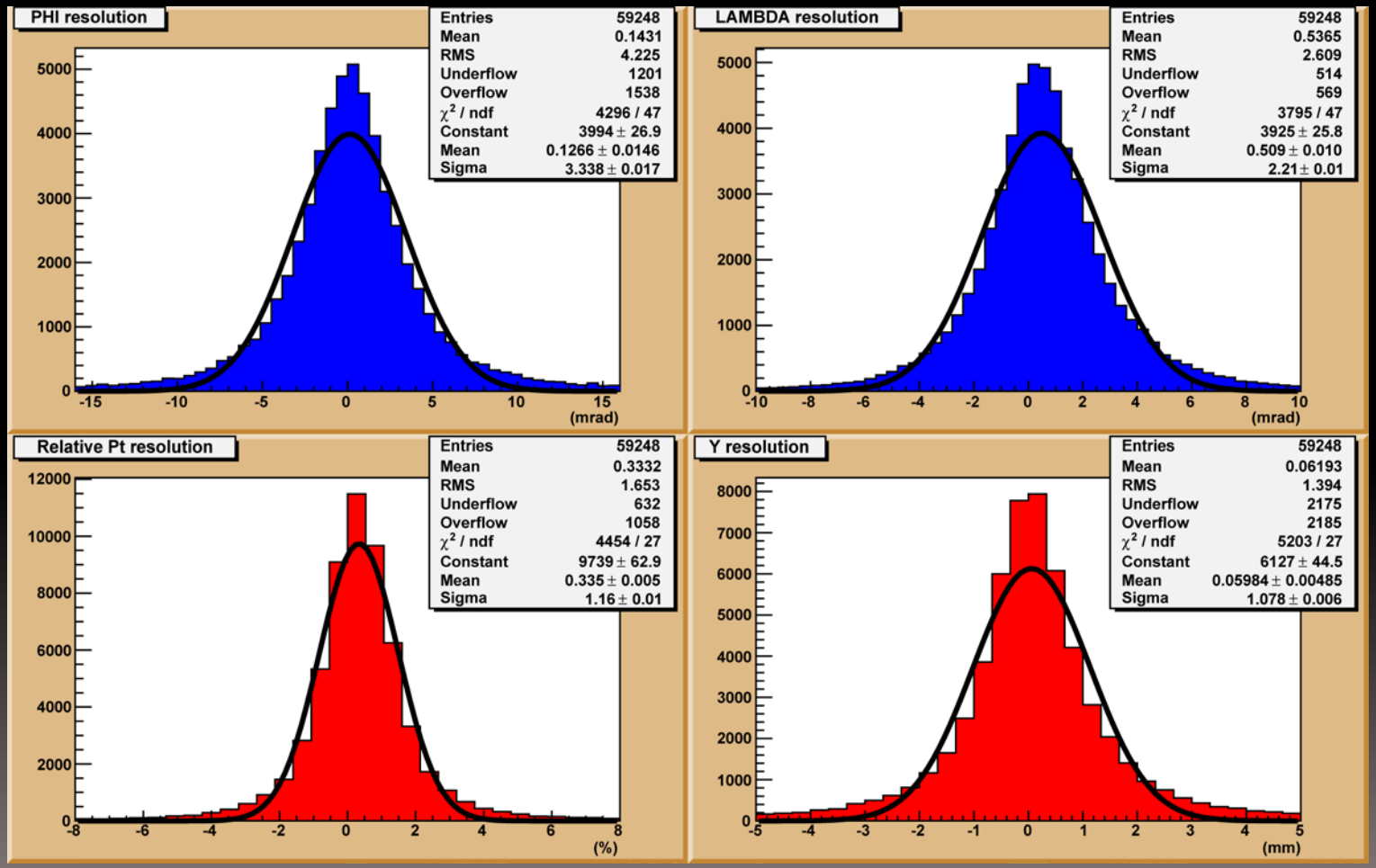
Global Tracking

- Momentum, Lambda, and Phi resolution increases.
- No difference between CPU and GPU version
- Y- and Z- resolution decreases slightly.

  → Currently under investigation.

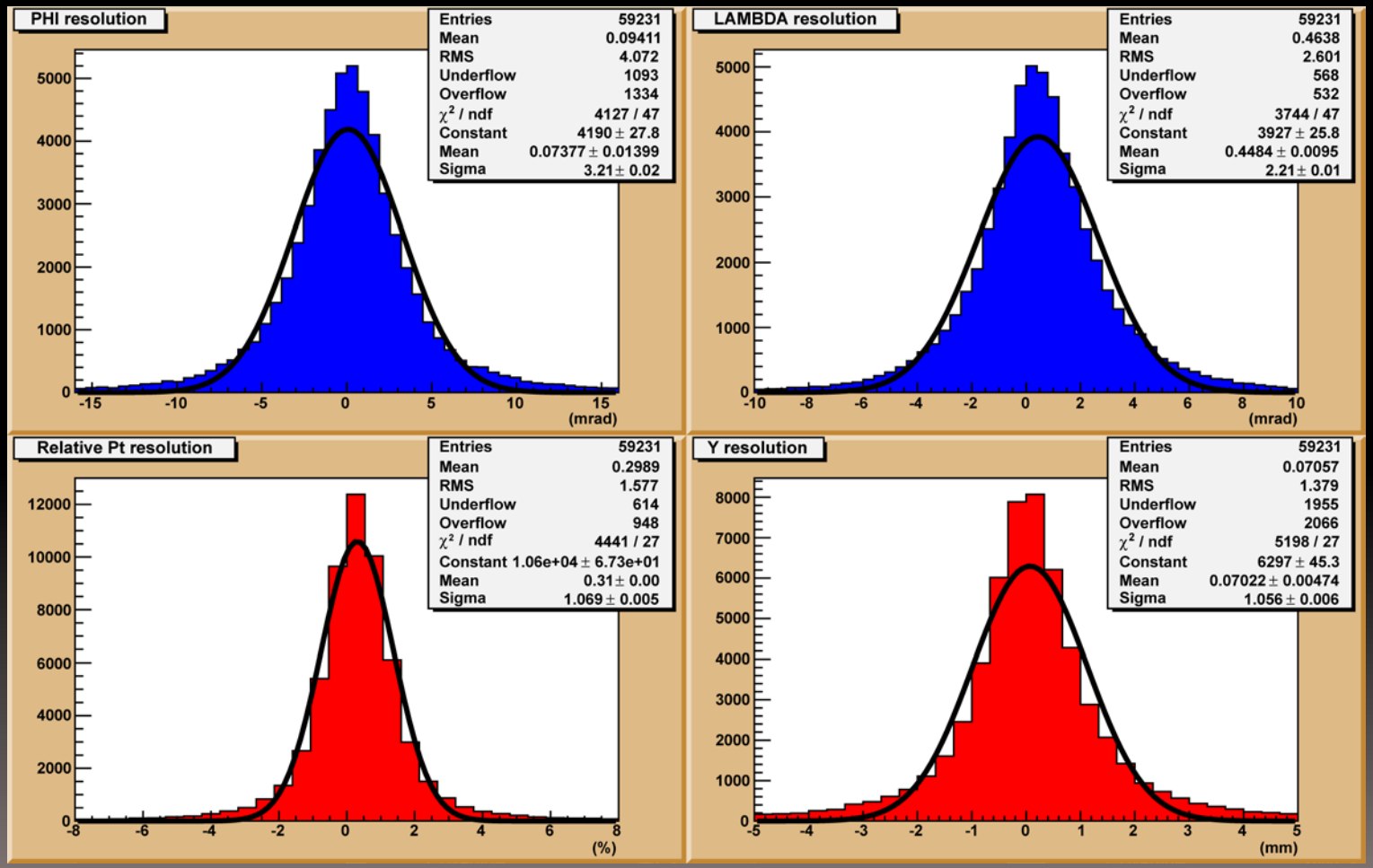
Global Tracking

Resolution without global tracking



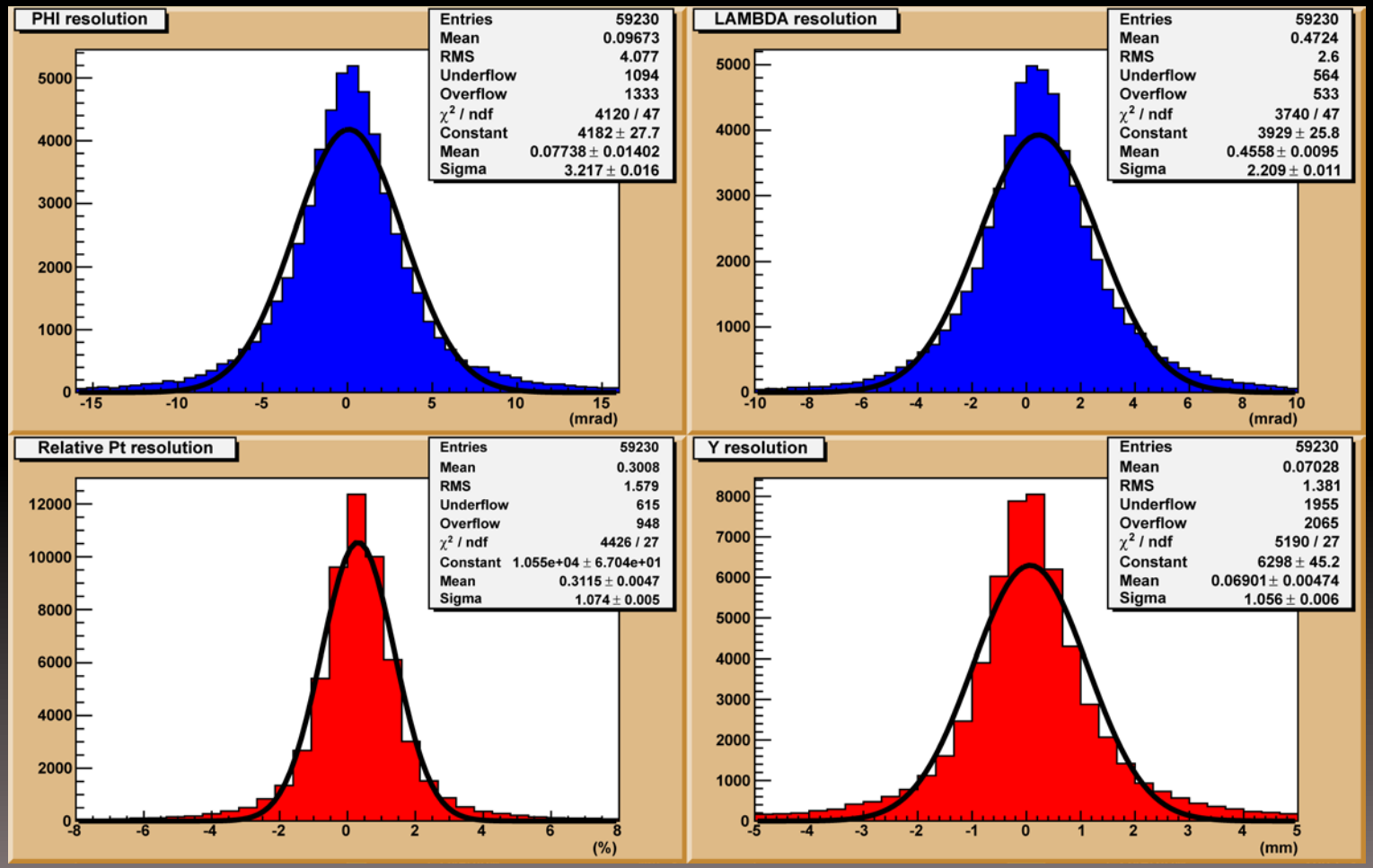
Global Tracking

Resolution with global tracking (on CPU)



Global Tracking

Resolution with global tracking (on GPU)



Global Tracking Performance

When activating global tracking, the tracking time increases by:

Tracker	Time increase
CPU Tracker (Single Threaded)	2.19%
CPU Tracker (Multi-Threaded)	12.60%
GPU Tracker	9.03%

Summary

- Three-fold performance increase of GPU tracker compared to six-core Nehalem CPU.
- GPU tracker performance does not depend on the CPU and depends linearly on input data size.
- Results of GPU and CPU tracker match almost completely. Only 0,00024% of the clusters differ due to non-associative floating point arithmetic.
- With the global tracking feature, the HLT GPU tracker can track accross slice boundaries.