

ALICE HLT High Speed Tracking and Vertexing

S. Gorbunov, K. Aamodt, T. Alt, H. Appelshäuser, A. Arend, B. Becker, S. Böttger, T. Breitner, H. Büsching, S. Chattopadhyay, J. Cleymans, I. Das, Ø. Djuvsland, H. Erdal, R. Fearick, Ø. S. Haaland, P. T. Hille, S. Kalcher, K. Kanaki, U. Keschull, I. Kisel, M. Kretz, C. Lara, S. Lindal, V. Lindenstruth, A. A. Masoodi, G. Øvrebek, R. Panse, J. Peschek, M. Ploskon, M. Richter, D. Rohr, D. Röhrich, B. Skaali, T. Steinbeck, A. Szostak, J. Thäder, T. Tveter, K. Ullaland, Z. Vilakazi, R. Weis, and P. Zelnicek for the ALICE collaboration

Abstract—The on-line event reconstruction in ALICE is performed by the High Level Trigger, which should process up to 2000 events per second in proton-proton collisions and up to 200 central events per second in heavy-ion collisions, corresponding to an input data stream of 30 GB/s.

In order to fulfil the time requirements, a fast on-line tracker has been developed which can optionally use GPU hardware accelerators. The algorithm combines a Cellular Automaton method being used for a fast pattern recognition and the Kalman Filter method for fit of found trajectories and for the final track selection.

A fast estimate of the vertex position is based on measurements from the Silicon Pixel Detector. The vertexer is used for the on-line monitoring of the ALICE interaction point.

I. INTRODUCTION

THE ALICE High-Level Trigger [1], [2] processes proton-proton collisions at 1kHz and heavy ion collisions at 300 Hz; with in average 25 tracks in each proton-proton event and up to 25000 tracks in the heavy ion events corresponding to an input data stream of 30 GB/s.

M. Richter, Ø. Djuvsland, H. Erdal, K. Kanaki, G. Øvrebek, Ø. S. Haaland, D. Röhrich, A. Szostak and K. Ullaland are with the Department of Physics and Technology, University of Bergen, Norway e-mail: Matthias.Richter@ift.uib.no

T. Alt, S. Gorbunov, S. Kalcher, V. Lindenstruth, T. M. Steinbeck, and J. Thäder are with the Frankfurt Institut für Informatik IfI, Frankfurt Institut für Advanced Studies FIAS, Ruth-Moufang-Str. 1, 60438 Frankfurt, Germany.

S. Böttger, T. Breitner, U. Keschull, M. Kretz, C. Lara, R. Panse, J. Peschek, D. Rohr, R. Weis, and P. Zelnicek are with the Kirchhoff Institute of Physics, University of Heidelberg, Germany

K. Aamodt, S. Lindal, B. Skaali, T. Tveter and are with Department of Physics, University of Oslo, Norway

M. Ploskon is with the Institut für Kernphysik, University of Frankfurt, Frankfurt am Main, Germany

H. Appelshäuser, A. Arend, H. Büsching are with the Institut für Kernphysik, Max-von-Laue-Str. 1, Goethe-Universität Frankfurt, 60438 Frankfurt am Main, Germany.

B. Becker, and Z. Vilakazi are with the iThemba LABS, University of Cape Town, PO Box 722, Somerset West 7129, South Africa.

S. Chattopadhyay and I. Das are with the High Energy Physics Division, Saha Institute of Nuclear Physics, 1/AF, Bidhan Nagar, Kolkatta 700 064, India.

J. Cleymans and R. Fearick are with the Department of Physics, University of Cape Town, Private bag X3, Rondebosch 7700, South Africa.

M. Ploskon is with the Physics Division, Lawrence Berkeley National Laboratory, 1 Cyclotron Road, 50R4049, Berkeley, CA 94720-8153, United States.

A. A. Masoodi is with the Department of Physics, Aligarh Muslim University, Aligarh 202001, U.P., India.

P. T. Hille is with the Department of Physics, Yale University, New Haven, CT 06520, United States.

I. Kisel is with the GSI Helmholtzzentrum für Schwerionenforschung GmbH, Planckstr. 1, 64291 Darmstadt, Germany.

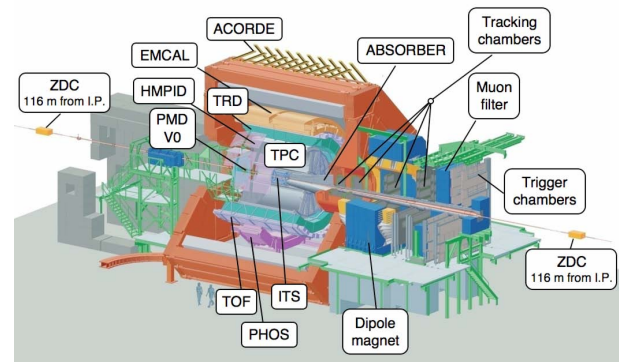


Fig. 1. The ALICE spectrometer at LHC.

Figures 2 and 3 show both types of events in order to demonstrate the complexity of the particle trajectories to be reconstructed in real-time.

The Time Projection Chamber (TPC) detector, which is shown on the figures 1, 2, 3, is the main tracking detector of the ALICE experiment. It consists of 18 sectors on either Z-side. The TPC detector measures the track positions in 159 rows as shown in figure 5.

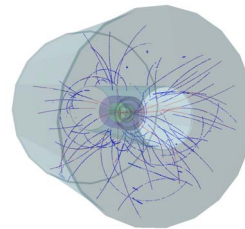


Fig. 2. Proton-proton event in the ALICE TPC detector. Real data, run 00010480 (2009).

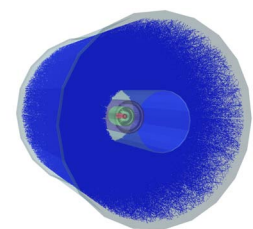


Fig. 3. Simulated heavy ion event in the ALICE TPC detector.

The overall reconstruction scheme is presented in figure 4. It starts with the TPC cluster finder, which finds the hits by identifying localized clusters and computing their centre of gravity. These reconstructed hits are sent to the sector tracker which reconstructs the tracks in each TPC sector individually. Then the sector tracks are merged by the track merger algorithm, and later updated with the measurements from the ITS detector. The reconstruction of the event's vertex and the physical triggers are running at the end of the reconstruction

tree structure. Typically every processing stage reduces the size of the event data. This scheme processes data as early as possible avoiding any unnecessary copy steps and uses all available data locality and parallelisation.

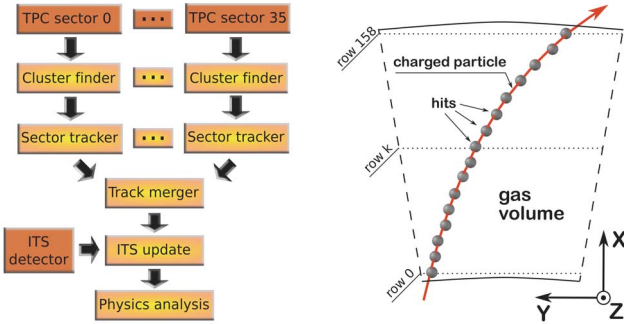


Fig. 4. HLT reconstruction scheme. Fig. 5. Geometry of a TPC sector.

The core of the event reconstruction happens in the TPC sector tracker, which creates the tracks from the measurements. It is the only component which processes the TPC hits, the higher level components operate on the reconstructed sector tracks.

II. TPC TRACKER ALGORITHM

An event coming from the detector only contains information about the spatial position of the hits, but no information about particles which caused the hits. The task of the track finder is to group the hits in such a way, that they form the particle trajectories.

This is a combinatorial pattern recognition problem. Since the potential number of hit combinations is enormous,¹ there is no exact solution of the problem, therefore heuristic methods are applied. Due to the large combinatorial background the key issue is the dependence of the reconstruction time on the number of tracks to be reconstructed. Figure 6 shows that the presented algorithm requires 130 us per track independent from the detector occupancy, thus the combinatorial part of the algorithm is built optimally.

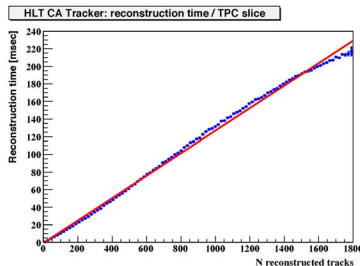


Fig. 6. TPC reconstruction time on CPU.

The track reconstruction algorithm starts with a combinatorial search for track candidates (tracklets), which is based on the Cellular Automaton method [3]. Local parts of trajectories

¹For example, given n tracks producing hits in each of 159 TPC rows, the number of possible hit combinations to create a single track is equal to n^{159} .

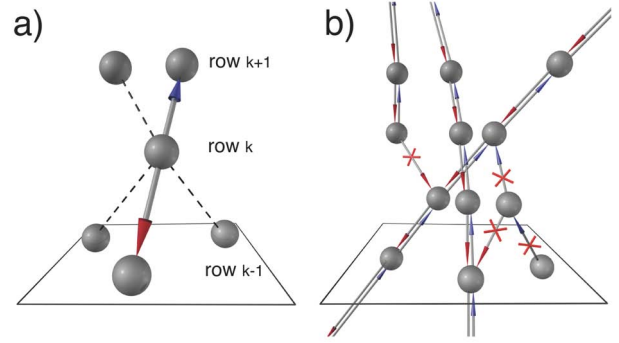


Fig. 7. a) Neighbours finder. b) Evolution step of the Cellular Automaton.

are created from geometrically nearby hits, thus eliminating unphysical hit combinations at the local level. The combinatorial processing composes the following two steps:

- 1. Neighbour finder: For each hit at a row k the best pair of neighbouring hits from rows $k+1$ and $k-1$ is found, as it is shown in fig. 7 a). The neighbour selection criteria requires the hit and its two best neighbours to form a straight line. The links to the best two neighbours are stored. Once the best pair of neighbours is found for each hit, the step is completed.
- 2. Evolution step: Reciprocal links are determined and saved, all the other links are removed (see fig. 7 b)).

Every saved one-to-one link defines a part of the trajectory between the two neighbouring hits. Chains of consecutive one-to-one links define the tracklets. One can see from fig. 7 b) that each hit can belong to only one tracklet because of the strong evolution criteria. This uncommon approach is possible due to the abundance of hits on every TPC track. Such a strong selection of tracklets results in a linear dependence of the processing time on the number of track candidates. When the tracklets are created, the sequential part of the reconstruction starts, implementing the following two steps:

- 3. Tracklet construction: The tracklets are created by following the hit-to-hit links as it is described above. The geometrical trajectories are fit using a Kalman Filter, with a χ^2 quality check. Each tracklet is extended in order to collect hits being close to its trajectory.
- 4. Tracklet selection: Some of the track candidates can have intersected parts. In this case the longest track is saved, the shortest removed. A final quality check is applied to the reconstructed tracks, including a cut on the minimal number of hits and a cut for low momentum.

The performance of the HLT track finder of 99.9 % for proton-proton events and 98.5 % for central Pb-Pb collisions has been verified on simulated events by using an official comparison macro. Corresponding efficiency plots are shown on figures 8 and 9. In addition to the high efficiency, the real-time reconstruction is an order of magnitude faster than off-line algorithm used as reference. The described algorithm has the advantage of a high degree of locality and allows for massively parallel implementation as outlined in the following section.

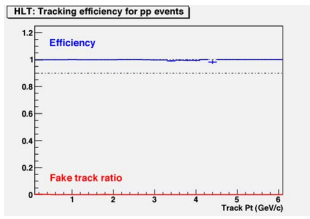


Fig. 8. Reconstruction performance for proton-proton collisions at 14TeV.

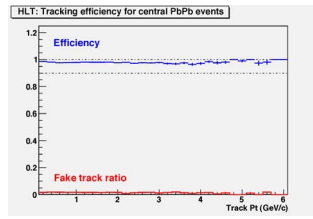


Fig. 9. Reconstruction performance for central heavy ion collisions at 5TeV.

There are many parts of the event reconstruction which are running after the tracker, in particular the primary vertex finder and the V0 finder. As it was noticed the HLT reconstruction was not only tested on simulated data, but it is running on the real data since 2009 [4], [5], [6]. A snapshot of one of the first ALICE proton-proton events, obtained by the HLT is shown on the figure 10.

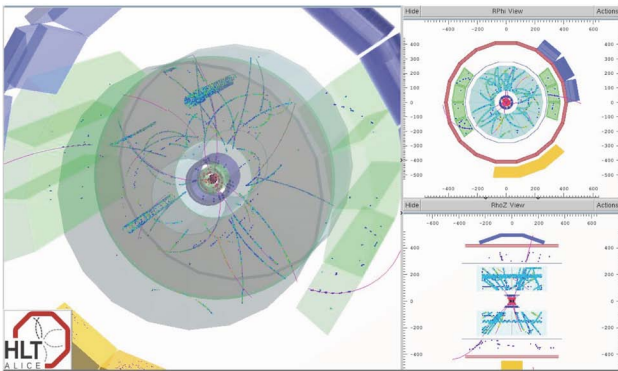


Fig. 10. The first proton-proton event, obtained by the ALICE High Level Trigger.

III. TRACKING ON GPU HARDWARE

In recent years the increase in processor clock speed stagnated but instead a trend to multi- and many-core chips came up. It is obvious, that for raw computation power, the best approach is a big set of small and simple cores as it has been realized within graphics cards for many years now. While at first they could only be used for very special problems using algorithms that had to be developed with a particular architecture in mind, today there are frameworks available to run general purpose code written in high level languages on GPUs with little changes.

All steps of the cellular Automaton tracker can be easily distributed on many independent processors. Primarily targeted at processing upcoming Pb-Pb events with more than 16,000 trajectories in the TPC and several million clusters (see fig.3), the tracker was adjusted to run on graphics processing units (GPUs), implementing 240 ALUs. A framework able to run the same source code on CPUs as well as GPUs was developed, where the same source files are included in wrappers for both processor types.

The required processing time for the various stages of the tracking algorithm (refer to the previous section) are shown

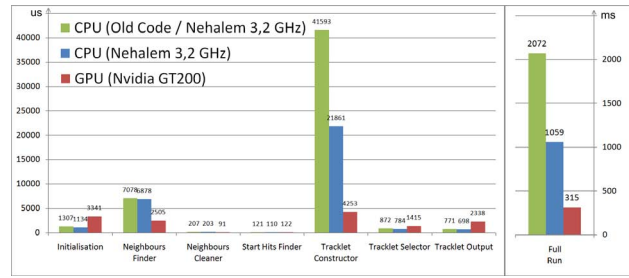


Fig. 11. CPU and GPU Tracker Performance of the named processing steps.

in fig. 11. Tracklet Construction (step 3) is the most time consuming part of the algorithm. GPU computing is organized in groups of 32 SIMD threads, called warps. High utilization of the GPU processing slots can be achieved only by introducing a custom scheduler, because all threads in one warp have to wait for the thread processing the longest tracklet. One thread extrapolates a tracklet only for a constant amount of rows. Afterwards all unfinished tracklets are redistributed among threads and multiprocessors. Further some pre-filtering is introduced to remove very short tracklets from the queue before even starting the extrapolation. For the scheduler to work efficiently the tracker algorithm has to process multiple sectors in parallel. This ensures to always have enough tracklets available for scheduling. A GPU utilization of almost 70 % was achieved, where canonical implementations resulted in only 20 % utilization.

The initialisation and output steps do not involve computation but scatter gather I/O. Therefore these processing steps remain on the host CPU.

The implementation of simultaneous processing of multiple sectors allows the use of a processing pipeline where data processing and data exchange between CPU and GPU overlap. After proper initialisation the CPU can immediately process the next slice while the GPU starts tracking the first one, as illustrated in fig. 12.

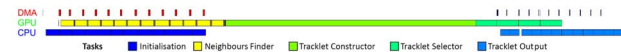


Fig. 12. Asynchronous GPU event processing.

While porting the tracker, the memory model was slightly changed. This resulted in more locality for optimal usage of available GPU memory bandwidth and also had positive cache effects on the CPU. The new tracker code performs better by a factor of two on modern CPUs (benchmarked using 3.2 GHz Intel Nehalem, 8 threads and data from Monte-Carlo simulation) while the GPU versions outperforms the Processor by another factor 3.3 for central lead lead collisions (fig. 12).

The GPU tracker is incorporated to the HLT framework and will run online in 2010.

IV. HIGH-SPEED VERTEXING IN ALICE HLT

The event vertex is reconstructed in the HLT by two complementary algorithms. The main vertexer is based on

the reconstructed tracks. It is applied at the end of the reconstruction chain and provides primary and V0 vertices.

In addition to the main finder, a fast estimate of the vertex position is provided using only measurements from the Silicon Pixel Detector (SPD). The SPD vertex finder is used for the on-line monitoring of the ALICE interaction point and supplements the main vertex finder.

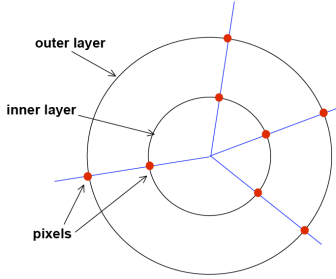


Fig. 13. SPD tracks.

The Silicon Pixel Detector (SPD), being the innermost ALICE detector, consist of two radial layers of silicon pixels placed at $\sim 4\text{cm}$ and $\sim 8\text{cm}$ respectively (see fig. 13). The detector provides pixel measurements (XYZ points). As the SPD provides only two measurements for each track, there is not enough redundancy for accurate reconstruction of the trajectories. Therefore any combination of inner and outer measurements is considered as a possible track (fig. 13). As the track curvature can not be reconstructed, the tracks are considered as straight lines. The event vertex is defined as a point of intersection of the highest number of tracks.

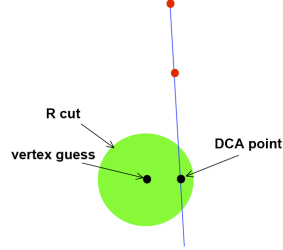


Fig. 14. Track selection in XY.

array. At the second step, the most dense area is found in the Z-array. Tracks which belongs to the found Z-area are considered as primary tracks.

At the last step of the algorithm, the vertex position is fitted by the LSM method as the closest point to all the selected tracks.

The SPD vertex finder has been tested on 1000 simulated 14TeV proton-proton events. It shows the speed of 3500 events per second and provides the vertex position with an accuracy of 250um in XY and 170um in Z directions respectively. An example of the reconstructed real data is shown on fig. 15.

V. SUMMARY

A fast on-line tracker has been developed for the ALICE High Level Trigger. The algorithm combines a Cellular Automaton method being used for a fast pattern recognition and the Kalman Filter method for fit of found trajectories and for the final track selection.

The on-line algorithm has proved its high performance on Monte-Carlo events (99.9% track finding efficiency for 14 TeV proton-proton events and 98.5% efficiency for central Pb-Pb events).

An important feature of the on-line tracking algorithm is the ability to use GPU hardware accelerators, giving another order of magnitude speedup for the data processing. The GPU tracker is integrated to the High Level Trigger framework and is used for the on-line event reconstruction in the ALICE experiment.

The event vertex is reconstructed by two complementary algorithms. The main HLT vertexer is based on the reconstructed tracks. It provides primary and V0 vertices and is applied at the end of the reconstruction chain. In addition to the main finder, a fast estimate of the vertex position is provided using only measurements from the Silicon Pixel Detector. The silicon vertexer processes up to 3500 events per second, providing the vertex position with an accuracy of 250um in XY and 170um in Z direction respectively. The algorithm is used for the on-line monitoring of the ALICE interaction point and supplements the main vertex finder.

The HLT software is running well since 2009 performing the full on-line event reconstruction.

REFERENCES

- [1] ALICE collaboration. ALICE - Technical Proposal for A Large Ion Collider Experiment at the CERN LHC. *CERN/LHCC 1995-71*, 1995.
- [2] K. Aamodt et al. The ALICE Collaboration. The ALICE Experiment at the CERN LHC. *JINST 3 S08002*, 2008. doi: [10.1088/1748-0221/3/08/S08002](https://doi.org/10.1088/1748-0221/3/08/S08002).
- [3] I. Kisel, Event reconstruction in the CBM experiment, *Nucl. Instr. and Meth A 566* (2006), pp. 8588. Event reconstruction in the CBM experiment. *Nucl. Instr. and Meth A 566*, 2006, pp.85-88.
- [4] K. Aamodt et al. (ALICE collaboration). First proton-proton collisions at the LHC as observed with the ALICE detector: measurement of the charged particle pseudorapidity density at $\sqrt{s} = 900$ GeV. *Eur.Phys.J. C*, 65:111–125, 2010. arXiv: [1004.3034](https://arxiv.org/abs/1004.3034).
- [5] J. Thäder et al. First Proton-Proton Collisions in the ALICE High-Level Trigger. *Proc.17th Real Time Conference, Lisbon 2010*
- [6] M. Richter et al. Event Reconstruction Performance of the ALICE High Level Trigger for p+p collisions. *Proc.17th Real Time Conference, Lisbon 2010*

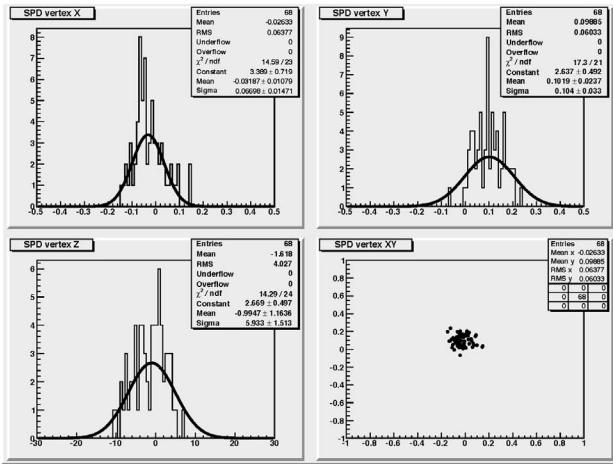


Fig. 15. Reconstructed HLT SPD vertex. Real data, run 00010480 (2009).

The vertex finder algorithm consist of three steps:

- selection of tracks in XY
- selection of tracks in Z
- fit of the vertex position

The track selection is performed separately in XY and Z projections. First, for each track its closest point to the vertex guess (DCA point) is estimated (see fig. 14). Tracks that do not pass a radial cut in XY are removed. For all the remaining tracks the Z-positions of their DCA points are stored in an