Online Event Reconstruction in the CBM Experiment at FAIR

Valentina Akishina[1,⋆] and *Ivan* Kisel[1,2,3,⋆⋆] for the CBM collaboration

[1] *Goethe-Universität, Frankfurt am Main, Germany*
[2] *FIAS, Frankfurt am Main, Germany*
[3] *GSI Helmholtz Center for Heavy Ion Research GmbH, Darmstadt, Germany*

Abstract. Targeting for rare observables, the CBM experiment will operate at high interaction rates of up to 10 MHz, which is unprecedented in heavy-ion experiments so far. It requires a novel free-streaming readout system and a new concept of data processing. The huge data rates of the CBM experiment will be reduced online to the recordable rate before saving the data to the mass storage. Full collision reconstruction and selection will be performed online in a dedicated processor farm. In order to make an efficient event selection online a clean sample of particles has to be provided by the reconstruction package called First Level Event Selection (FLES).

The FLES reconstruction and selection package consists of several modules: track finding, track fitting, event building, short-lived particles finding, and event selection. Since detector measurements contain also time information, the event building is done at all stages of the reconstruction process. The input data are distributed within the FLES farm in a form of time-slices. A time-slice is reconstructed in parallel between processor cores. After all tracks of the whole time-slice are found and fitted, they are collected into clusters of tracks originated from common primary vertices, which then are fitted, thus identifying the interaction points. Secondary tracks are associated with primary vertices according to their estimated production time. After that short-lived particles are found and the full event building process is finished. The last stage of the FLES package is a selection of events according to the requested trigger signatures. The event reconstruction procedure and the results of its application to simulated collisions in the CBM detector setup are presented and discussed in detail.

1 Introduction

One of the main objectives of the physics program for the future heavy-ion experiment CBM (FAIR, Germany) [1] is to understand the properties of strongly interacting matter at very high baryonic densities. The physics program of CBM includes the measurement of yields, spectral distributions, correlations and fluctuations of the particles produced in the heavy-ion reactions. The CBM detector setup is designed to measure not only bulk hadrons, but very rare probes like multi-strange hyperons or charmed hadrons as well. In order to measure rare probes with sufficient statistics the experiment will operate at high interaction rates of up to 10 MHz, which puts strong constraints on the data

[⋆] e-mail: V.Akishina@gsi.de
[⋆⋆] e-mail: I.Kisel@compeng.uni-frankfurt.de

processing. It requires very fast and radiation-hard detectors, and a novel data read-out and analysis concept based on free streaming front-end electronics and a high-performance computing cluster for online event selection. In order to make an efficient event selection online a clean sample of particles has to be provided by the reconstruction package called First Level Event Selection (FLES) [2].

2 Reconstruction of free-streaming data in the CBM experiment

The huge data rates force the CBM experiment to reduce data online before saving it to the mass storage. In order to reduce data in real time, the experiment has to implement triggers. Simple hardware triggers typical for traditional latency-limited trigger architectures are inapplicable for the case of CBM, since the experiment is targeted to events with complicated patterns, which require full event reconstruction. Therefore, CBM plans to implement sophisticated software trigger, which will process raw detector data, reconstruct collision patterns online and take a trigger decision based on the reconstructed information.

To implement the described scheme CBM will benefit from a new data acquisition concept with autonomous, self-triggered front-end electronics. Every detector channel in this case will deliver time-stamped data signals upon its activation. These messages will be transported to the readout buffer, the size of which will be adjusted to the power of a large computer farm, where it will be reconstructed.

The detector data obtained during a certain period of time is called a time-slice. As a consequence of the read-out concept, the detector signals inside a time-slice are not associated with physical collisions. However, in order to obtain correct physics interpretations, one needs to extract information on the individual collisions of ions from the continuous time-slice data. Thus, the reconstruction routines of CBM should deliver event-associated data in the end. The process of event-association is called event building and should be a part of the standard reconstruction chain.

The event building is not a trivial task for CBM, especially for the cases of the highest iteration rates, when information on different collisions overlaps in time. This case requires sophisticated algorithms, which will allow to reliably resolve different collisions from each other, using not only time, but also spacial information. This is a task for a FLES reconstruction package.

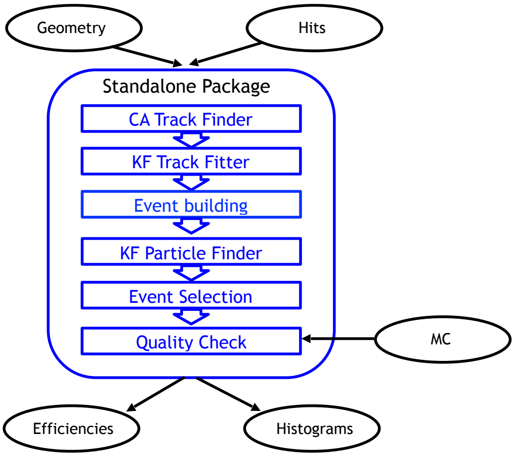

Figure 1. Block diagram of the FLES package.

The FLES package consists of several modules (see figure 1): track finder, track fitter, event builder, particle finder and physics selection. As an input the FLES package receives a simplified

Table 1. Track reconstruction efficiency performance for one hundred minimum bias AuAu UrQMD collisions at 25 *A*GeV for the event-by-event reconstruction and the time-slice-based reconstruction at 0.1 MHz, 1 MHz and 10 MHz interaction rates, presented for different track categories.

Track category, %	E-by-E	10^5Hz	10^6Hz	10^7Hz
All tracks	92.1	92.6	92.6	92.2
Primary high-p	97.9	98.2	98.2	97.9
Primary low-p	93.6	94.1	94.1	93.5
Secondary high-p	92.0	92.7	92.7	92.0
Secondary low-p	65.7	66.7	66.6	65.9
Clone level	2.8	0.3	0.3	3.1
Ghost level	4.9	3.5	3.5	4.2
MC tracks found	145	146	146	145

geometry of the tracking detectors and the hits, which are created by the charged particles crossing the detectors. Tracks of the charged particles are reconstructed by the Cellular Automaton (CA) track finder [3]. The Kalman filter (KF) based track fit [4] is used for precise estimation of the track parameters. The event builder algorithm takes as input fitted tracks and based on this information builds events via splitting tracks into event-corresponding groups. The next step of the data processing chain is particle reconstruction. The short-lived particles, which decay before the tracking detectors, can be reconstructed via their decay products only. The KF Particle Finder [5], which is based on the KF Particle package [5, 6] is used in order to find and reconstruct the parameters of short-lived particles by combining already found tracks of the long-lived charged particles. The algorithm takes as input events delivered by the event-builder algorithm. The KF Particle Finder also selects particle-candidates from a large number of random combinations. In addition, a quality check module is implemented, that allows to monitor and control the reconstruction process at all stages.

The FLES package in the CBM experiment will be executed for the online selection and the off-line analysis on a dedicated many-core CPU/GPU farm. The package is platform and operating system independent. The farm is currently estimated to have a compute power equivalent to 60 000 modern CPU cores. The FLES algorithms have to be therefore local and parallel with respect to data and use the full potential of modern and future many-core CPU/GPU architectures.

3 4D Cellular Automaton track finder

The track finder algorithm used to reconstruct tracks in the tracking detectors of CBM is based on the Cellular Automaton method [7]. The CA track finder takes as an input hit measurements from the Silicon Tracking System (STS detector) in the form of a time-slice, which includes time and spacial measurements. The track finding procedure starts with combining the hits, delivered by the reconstruction algorithms of STS detector, into triplets – combination of three hits on adjacent stations. The triplet structure was chosen, since it allows to estimate the momentum of particle, which could produce it. The triplets with two common hits are combined into track candidates. In order to take into account detector inefficiency, a missing hit in one station is tolerated for a reconstructed track. The track candidates should survive a dedicated selection based on the track length and calculated χ^2-value to be accepted to the reconstructed tracks.

In order to optimize and speed up the process of track reconstruction the track finding procedure is performed in several iterations [8]. At first the track finder algorithm is searching for the tracks, which are easier to reconstruct, for instance, high momentum and primary tracks. Having found these tracks, the algorithm removes hits, which were used to construct these tracks, from the input information. Thus, the combinatorics, which is higher for the iterations when searching for low-momenta and secondary tracks gets significantly reduced. Therefore, in the first iteration only high-momentum primary tracks are reconstructed, in the second one – low-momentum primary tracks, and then – all other tracks.

Input time information is used in the algorithm to the same extent and in similar manner as it is done with the spacial coordinates. The same logic is used while constructing triplets: the hits in the triplet should belong to the same particle, therefore they should correlate not only in space, but also in time. Since the estimated time of flight between two consecutive stations is negligible in comparison to the detector time precision, the hits, belonging to the same track, should coincide within three σ-s of the detector time precision.

In order to estimate the performance of the algorithm, the efficiency calculation tests based on the generated Monte-Carlo information for one hundred AuAu minimum bias UrQMD [9, 10] events at 25*A* GeV have been performed on a server with Intel Xeon E7-4860 CPUs. For evaluation purposes the particle is called reconstructable if it has crossed at least 4 stations of the STS detector. A reconstructed track is assigned to a simulated particle, if at least 70% of its hits have been caused by the same particle. If a certain particle is assigned to more than one track, such additionally reconstructed tracks are regarded as clones. If a track was not assigned to any particle it is called a ghost. The resulting track reconstruction efficiencies for the cases of event-by-event analysis (so-called 3D analysis) as well as for the 4D case (with included time measurement, as well as 3-dimensional spacial information) while reconstructing time-slices, produced out of one hundred AuAu minimum bias UrQMD events at 25*A* GeV, are presented in table 1 [8].

The inclusion of the time measurement have made it possible to achieve an efficiency performance comparable to the case of the event-by-event analysis. Moreover, the track reconstruction efficiency has been improved after taking into account the STS time measurement in comparison to the event-based performance. The effect of efficiency improvement is present even for the extreme case of 10 MHz interaction rate. It can be explained by the presence of slow particles, which create random combinations of hits in the 3D case. These random combinations can be rejected in the case of time-slices thanks to the hit time measurement cut, thus improving the performance [8].

The algorithm has been included into the CBMROOT framework [11]. The simulation of detector response in the framework provides a time measurement, taking into account the anticipated behavior of the detector. The performance for the algorithm included into CBMROOT framework is presented in table 1. It is comparable to the 3D case.

The CA track finder is both vectorized (using SIMD instructions) and parallelized (between CPU cores) with the help of OpenMP interfaces [12]. The algorithm shows strong scalability on many-core systems. A speed-up factor of 10.6 was achieved on an Intel Xeon E7-4860 CPU with 10 hyper-threaded physical cores [13].

4 Kalman filter track fit

The Kalman filter is intensively used in the CA track finder for obtaining parameters for partially reconstructed track segments, therefore its fast implementation on modern CPU/GPU computer systems and stability in single precision are crucial not only for the fitting procedure, but for the track finder as well. A massive optimization was performed with the initial purpose of vectorisation of the algorithm [2]. To optimize the memory usage, a magnetic field approximation was implemented for particle propagation instead of using the full magnetic field map, which takes about 70 MB and therefore does not fit into the CPU cache memory. The magnetic field is approximated with a polynomial function of fifth order at each detector station. During the fit of a track the field behavior between the stations is approximated with a parabola taking field values at the three closest measurements along the track. The first measurement is processed with a dedicated procedure, which increases the

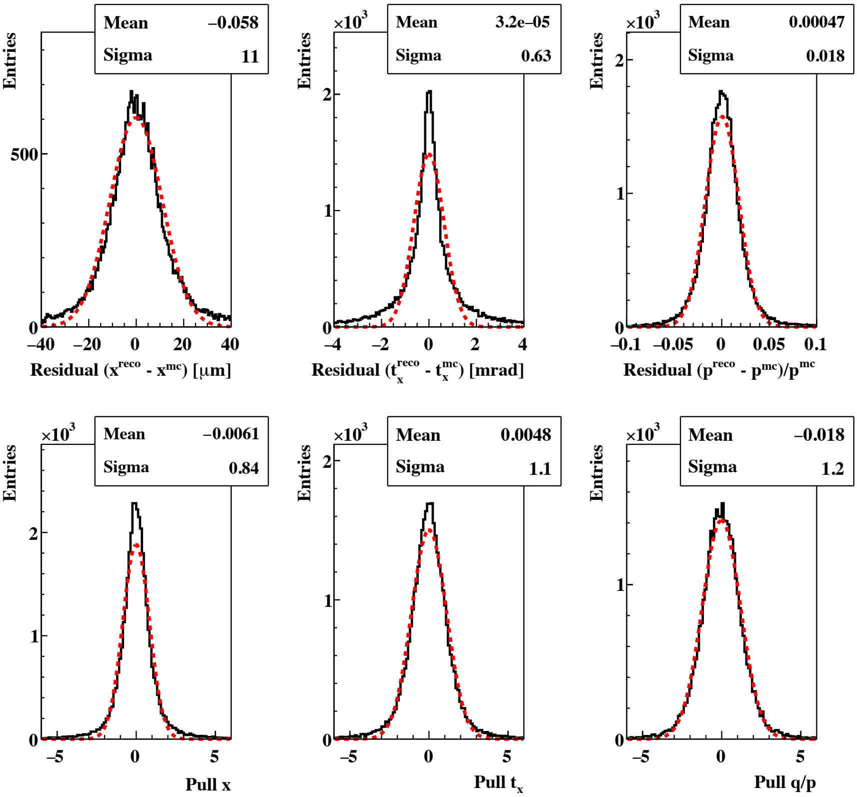

Figure 2. Residuals and pulls distributions for the x, t_x and q/p track parameters, reconstructed with 4D CA track finder and Kalman Filter track fit. The dashed lines correspond to Gaussian fits of the distributions.

numerical stability of the method in single precision. The track propagation in the non-homogeneous magnetic field is done by an analytic formula based on the Taylor expansion [14]. The analytic formula allows to obtain the same track fit quality as the standard fourth order Runge-Kutta method, while being 40% faster. Operator overloading has been used to keep flexibility of the algorithm with respect to different CPU/GPU architectures. All these changes have increased the processing speed of the SIMD KF track fit algorithm down to 1 μs per track. This is an improvement by a factor 10000 with respect to the original scalar version of the algorithm [4].

Since CBM is a forward geometry experiment, the following track parameters, which are measured at a certain z position, are used: x and y coordinates, t_x and t_y are the track slopes in the XZ- and YZ-planes, q/p is the inverse particle momentum with the sign of the particle charge and t is the time. The time parameter is fully included in the fitting procedure and treated the same way as the spacial parameters. The obtained fitted time parameter is particularly important and intensively used during the next element of reconstruction chain – event builder. In order to demonstrate the fit quality the residuals and pulls of the parameters in the track state vector have been calculated (see figure 2). The residuals of the track parameters are defined as differences between the reconstructed parameters and the true Monte-Carlo values. The pulls are defined as the residuals normalized by the estimated errors of the track parameters. In the ideal case these should be unbiased and Gaussian distributed with width of 1.0.

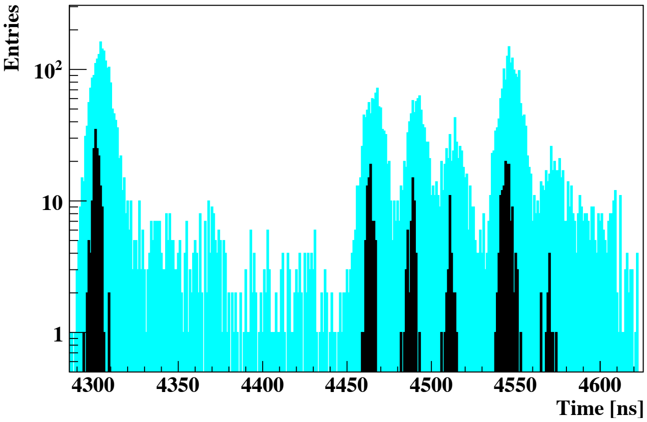

Figure 3. The distribution of the STS hit time measurement in the time-slice with one hundred minimum bias AuAu UrQMD events at 25 *A*GeV at 10 MHz is shown with a light blue color, the distribution of the fitted track time is shown with black color: the reconstructed tracks are clearly grouped into the clusters of events (color online).

5 Event builder

The event building algorithm based on found tracks will deliver reconstructed events for the case of the highest CBM interaction rate of 10 MHz. The first version of the algorithm is implemented based on the information from the STS detector alone and will be extended in future for other sub-detector systems. For instance, the Time-of-Flight (ToF) detector, the time resolution of which is several orders of magnitude better than the STS precision, is expected to drastically improve the ability to resolve the events, which contain ToF information [8].

The track-based event builder splits the reconstructed tracks into event-corresponding groups based on the fitted time parameter, extrapolated to the region of primary vertex, and the estimated errors, obtained after the Kalman Filter track fit [15]. The algorithm starts grouping tracks around the best fitted track with the smallest estimated time errors. All tracks which coincide in the reconstructed time within 3 σ of estimated time errors are assigned to the same event. The procedure is repeated for all tracks until none of them are left.

The initial distribution of hits measurements illustrating the complexity of defining event borders in a time-slice at the interaction rate of 10 MHz is shown in figure 3 in a light blue (light gray in print). This is a typical example of the fact that at such an extreme interaction rate there are no isolated events. After the 4D time-slice-based reconstruction we assign to each track a time measurement, which is calculated as the average of its hits measurements. As a next step we need to group the reconstructed tracks in time into clusters of tracks belonging to same collision, identifying thus physical events. The resulting time distribution of track clusters is shown in a black color in figure 3. The reconstructed track clusters represent groups corresponding to collisions the tracks of which originate from. Even in the area of the severe initial event overlap on a hit measurement level the time-based CA track finder allows to resolve tracks from different collisions in time.

The tests performed with one hundred AuAu minimum bias UrQMD events at 25*A* GeV have shown that 83% of events were reconstructed without any event merging, 17% of events were reconstructed in merged double event clusters. For evaluation purpose an event is called merged if at least one track from the reconstructed event was assigned to a wrong event. Absence of event splitting was achieved. Due to properties of Poisson distribution, which determines the time distance between

Table 2. The reconstruction efficiency for minimum bias AuAu UrQMD collisions at 10 *A*GeV in the cases of event-by-event reconstruction and time-slice-based reconstruction at 10 MHz interaction rate for K_s^o, Λ and Ξ^-.

Particle/ case	3D, %	10 MHz, %
K_s^o	22.9	21.2
Λ	21.9	19.6
Ξ^-	7.8	6.3

consecutive collisions, the method based on time information alone can not guarantee total resolution of collisions since the probability of events coinciding in time within time precision is never zero. Therefore, for the extreme scenario of interaction rate of 10 MHz the remaining 17% of events are to be resolved later with the use of the multi-primary-vertex analysis and the full event topology.

6 KF Particle Finder

The next step in the FLES data processing chain is the reconstruction of particles. It is done in two stages. First, particles registered in the CBM detector system are reconstructed. Then short-lived particles decayed before or inside the setup are searched based on both their charged and neutral daughter particles.

The KF Particle Finder package [2, 5] is used in CBM to reconstruct short-lived particles. The event builder algorithm provides a set of tracks grouped into the event to the KF Particle Finder as an input, thus, allowing the physics analysis for the time-stamped data. Since the FLES package is developed to run on many-core computer architectures, the reconstruction of particles is done in parallel that provides a possibility for a global competition between particle candidates. Such a global event reconstruction significantly improves suppression of a combinatorial background and provides for further physics analysis a very clean sample of particles produced at different stages of heavy-ion collision. Recently, the package has been extended to hyper-nuclei and weak decays involving neutral particles in the final state.

The resulting reconstruction efficiencies normalised to the total number of produced particles are given in table 2. The resulting efficiency of short-lived particle reconstruction for the 10 MHz interaction rate is similar to the event-based analysis. The loss of particle reconstruction efficiency despite stable track reconstruction efficiency, which is normalised to the number of reconstructable tracks, for the case of time-based reconstruction can be explained by the smaller number of produced hits due to the strip dead time.

The search for the short-lived particles is done in one go, thus minimizing access to the memory. Together with the optimization and vectorization of the code this allows to achieve a high speed even in the presence of a huge combinatorics. The speed of the KF Particle Finder per core on a server with Intel Xeon E7-4860 CPUs is 1.4 ms per minimum bias AuAu UrQMD collision and 10.5 ms per central AuAu UrQMD collision at 25 *A*GeV.

7 Conclusions

The reconstruction chain of CBM being developed enables the free-streaming data processing by dedicated time-based algorithms. It allows to reconstruct individual events out of continuous time-based data. The reconstructed events, each of which in the ideal case corresponds to one real interaction in the target, serve as input for high-level physics analysis. The problem of overlapping events at highest interaction rates will be addressed by studying the possibility to reconstruct multiple primary vertices inside one event.

The FLES package for the CBM experiment contains all reconstruction stages: track finding, track fitting, short-lived particles finding, event building and event selection. Time-based 4D Cellular Automaton and the Kalman filter algorithms are used for finding and fitting tracks both in space and time, that allows to achieve a high track reconstruction efficiency up to 92% and the track parameters quality with about 1.5% of the momentum resolution.

The 4D event building has been implemented for the CBM experiment. It allows to resolve the major part of overlapping on a hit level events and group tracks into event-corresponding clusters without event splitting. About 17% of events are merged and can not be separated using only the time information at the interaction rate of 10 MHz. The first version of the algorithm is implemented and will be enhanced with multi-vertex reconstruction and information from the TOF detector.

The KF Particle Finder package shows a high reconstruction efficiency with an optimal signal to background ratio. The physics simulations and performance studies were extended to hyper-nuclei and weak decays involving neutral particles in the final state.

Thus, the first version of the full reconstruction chain for the CBM experiment is established. All the algorithms are developed in a simple and general form to be experiment specifics independent with efficient use of many-core architectures. The package is vectorized using SIMD instructions and parallelized between CPU cores. All algorithms are optimized with respect to the memory usage and speed.

Acknowledgements

This work was supported by HICforFAIR, FIAS and BMBF.

References

[1] B. Friman, C. Hohne, J. Knoll, S. Leupold, J. Randrup, R. Rapp, and P. Senger, *The CBM physics book: Compressed baryonic matter in laboratory experiments* (Lect. Notes Phys. 814, 2011) 980

[2] I. Kisel, I. Kulakov, and M. Zyzak, IEEE Transactions on Nuclear Science **60**, 5, 3703 (2013)

[3] V. Akishina, *4D Event Reconstruction in the CBM Experiment* (Dissertation thesis, Goethe university, Frankfurt am Main, 2017) 181 pp.

[4] S. Gorbunov, U. Kebschull, I. Kisel, V. Lindenstruth, and W.F.J. Müller, Computer Physics Communications **178**, 5, 374–383 (2008)

[5] M. Zyzak, *Online Selection of Short-Lived Particles on Many-Core Computer Architectures in the CBM Experiment at FAIR* (Dissertation thesis, Goethe university, Frankfurt am Main, 2016) 159 pp.

[6] S. Gorbunov, *On-line reconstruction algorithms for the CBM and ALICE experiments* (Dissertation thesis, Goethe University of Frankfurt, 2012) 104 pp.

[7] I. Kisel, Nucl. Instr. and Meth. **A566**, 85–88 (2005)

[8] V. Akishina, I. Kisel, I. Vassiliev, and M. Zyzak "Time-based Reconstruction of Free-streaming Data in CBM," this journal

[9] S. A. Bass, *et al.*, Prog. Part. Nucl. Phys. **41** (1998)

[10] M. Bleicher, *et al.*, J. Phys. G: Nucl. Part. Phys. **25** (1999)

[11] CBMROOT framework: https://redmine.cbm.gsi.de/projects/cbmroot

[12] The OpenMP API specification for parallel programming http://openmp.org/wp

[13] V. Akishina and I. Kisel, J. Phys. Conf. Ser. **599**, 1, 012024 (2015)

[14] S. Gorbunov and I. Kisel, Nucl. Instr. and Meth. **A559**, 148–152 (2006)

[15] V. Akishina and I. Kisel, IEEE Trans. Nucl. Sci. **62**, 6, 3172 (2015)