UDC 519.254; 539.1

Fast Global Tracking for the CBM Experiment at FAIR

A. Lebedev*, C. Höhne†, I. Kisel†, G. Ososkov*

* *Laboratory of Information Technologies*
Joint Institute for Nuclear Research
Joliot-Curie 6, 141980 Dubna, Moscow region, Russia
† *GSI Helmholtzzentrum für Schwerionenforschung GmbH*
64291 Darmstadt, Germany

Particle trajectory recognition is an important and challenging task in the Compressed Baryonic Matter (CBM) experiment at the future FAIR accelerator at Darmstadt. In this contribution, the status of the global track reconstruction software for the CBM experiment is presented. The global track reconstruction procedure is based on track following and Kalman Filter methods. The track reconstruction efficiency for central Au+Au collisions at 25 AGeV beam energy using events from the UrQMD model is at the level of 93-95%. Since CBM has to process terabytes of input data produced at high collision rates, it is extemly important to develop fast track reconstruction algorithms. Possibilities to speed up the algorithms have been studied. A significant optimization of memory consumption and necessary combinatorics has been done. The usage of multithreading results in further acceleration. Overall, a factor 20 in speed could be achieved by these improvements.

Key words and phrases: high energy physics, CBM experiment, track reconstruction, Kalman Filter, track fit.

1. Introduction

The Compressed Baryonic Matter (CBM) [1] experiment is a heavy-ion fixed target detector at the future FAIR accelerator at Darmstadt. It is being designed for the investigation of the properties of highly compressed baryonic matter [2]. High reaction rates (up to 10 MHz), events with large track multiplicity (about 800 charged particles per central event in the CBM detector acceptance) and high hit density are expected at CBM. This leads to special requirements for the tracking software, which has to perform fast and stable in such environment. The CBM setup consists of several detectors including as tracking detectors the silicon tracking system (STS), located inside a large acceptance dipole magnet for track and vertex reconstruction and momentum determination; the muon detector (MUCH) or alternatively a set of Transition Radiation Detectors (TRD). Finally, hits from the Time of Flight (TOF) detector are attached to global tracks. The current design of the muon system consists of 5-6 hadron absorber layers made of iron of variable thickness, and of 2-3 tracking detectors in between each of the gaps. For the measurement of muons from low-mass vector mesons (ρ, ω, ϕ), the total iron absorber thickness is 125 cm (7.5 λ_I, where λ_I is the nuclear interaction length), whereas for muons from charmonia, 1 m of iron is added (total thickness of 13.4 λ_I). The currently forseen layout of the TRD consists of 12 identical layers, grouped in 3 stations. Each layer is formed of a radiator in front of gas detector. Stations are placed at 5 m, 7 m and 9 m downstream the target. The detailed layout of the detectors is still under discussion. For the TRD and the MUCH stations in the high track density region pad layout is forseen based on MWPC or GEM chamber technology. For the outer detector stations in MUCH where track densities are low, straw tube chambers are under discussion.

Received 28th November, 2009.
For the CBM collaboration.

2. Track Reconstruction

Global track reconstruction in CBM is based on track following using reconstructed tracks in the STS as seeds. In the STS track reconstruction is based on the cellular automaton method [3] and provides initial track parameters as starting point for the following track prolongation. This track following is based on the standard Kalman filter technique as proposed in [4] and is used for the estimation of track parameters and global trajectory recognition. Tracks are prolongated subsequently from one detector station to the next adding additional hits in each detector. Main logical components are track propagation, track finding, track fitting and finally a selection of good tracks. Each of the steps will be described in the following in some more detail.

2.1. Track Propagation

Track propagation is the main part of this track reconstruction method. The algorithm calculates the average trajectory and its errors in a covariance matrix while taking into account physics processes which may change the trajectory, i.e. energy loss, multiple scattering and also the influence of a magnetic field.

The propagation of the further trajectory is done according to the equation of motion. If the track passes a field free region a straight line is used for propagation and the transport matrix calculation. If passing a magnetic field the equation of motion for a charged particle is solved applying the 4^{th} order Runge-Kutta method. The transport matrix is calculated by integrating the derivatives.

In order to include the influence of the material along the trajectory, a geometry navigation routine based on the ROOT geometry package [5] searches for intersections of the prolongated track with detectors or passive material.

The influence of the material on the track momentum is taken into account by calculating the expected average energy loss due to ionization (Bethe-Bloch formula), bremsstrahlung (Bethe-Heitler formula) and direct pair production (estimated using table values). The influence on the error, i.e. the covariance matrix due to multiple scattering is included by adding process noise in the track propagation. Here, a gaussian approximation using the Highland formula is used to estimate the average scattering angle.

A detailed description of the developed track propagation can be found in [6].

2.2. Track Finding

In the track finding algorithm hits are attached to the propagated track at each detector station using three different methods. Either just the nearest hit is attached to the track (nearest neighbor algorithm), or a weighting method including all surrounding hits is used (weighting algorithm), or all hits within a certain environment are included by creating track branches (branching algorithm). For the first two methods, only one track is further propagated, the last method allows for several track branches to be followed, one for each attached hit. Common techniques to these algorithms are the track following using the Kalman Filter and the calculation of the validation gate for hits.

The assignment of new hits is done step by step at each detector station. After the track propagation to the next station possible hits are attached and track parameters are updated by the Kalman Filter. For the attachment of hits a validation gate is calculated in order to allow for a high degree of confidence in the hit-to-track assignment. In the context of Kalman-based tracking filters, a validation gate can be expressed as $v = rR^{-1}r^T < d$, where r and R are the residual vector and its covariance matrix, and d the chosen probability for rejecting the correct hit. The algorithm allows for missing hits due to detector inefficiencies.

The three different methods which can be chosen for hit-to-track assignment differ in the way how a situation is dealt with in which several hits lie inside the validation gate. With the branching method, a new track branch is created for each hit lying within the validation gate. Since the number of branches can grow very fast, the χ^2

value is calculated for each track branch and unlikely ones are rejected. For the other two methods no track branches are created. The nearest neighbor algorithm attaches the hit with the smallest v, if lying in the validation region at all. In the weighting algorithm all hits within the validation gates of all stations are assigned to the track. Then an iterative hit selection and track fitting procedure starts. In this procedure weights are assigned to each hit in the validation gate and a weighted mean of these hits is calculated. The weight is proportional to the multivariate Gaussian function and includes a so called temperature parameter T. The method uses simulated annealing to recalculate those weights in each iteration. Finally, the hit with the highest weight from each station is selected.

After track finding, so called clone tracks (consisting of a very similar set of hits) and ghost tracks (consisting of a random set of hits) have to be rejected while keeping correctly found tracks with high efficiency. The selection algorithm works in two steps. First, tracks are sorted by their quality which is defined by the track length and χ^2. Then, starting from the highest quality tracks all hits belonging to the track are checked. In particular, the number of hits shared with other tracks is calculated and the track is rejected if more than 15% hits are shared.

3. Speedup of the Algorithm

Since CBM has to process terabytes of input data produced at high collision rates, it is extremly important to develop fast track reconstruction algorithms. Possibilities to speed up the algorithms have been studied. First, the necessary combinatorics was optimized. As the evolution of hardware makes parallel programming more and more mainstream, a parallelization of the nearest neighbour tracking algorithm has been investigated.

The hit-to-track assignment procedure is a key routine in the track finding algorithms. The simplest way is to loop over all hits in the detector station, update track parameters with the Kalman filter method, calculate the validation gate and check whether the hit falls into the validation gate or not. However, these loops are very time consuming and moreover most of them without success as no hit is attached to the track. To solve this issue a fast search of hit-candidates has been developed. The method is executed before the described loop and selects a certain range of hits for the more precise check performed afterwards. The usage of this procedure in the track reconstruction results in 2 times less computational time (from 6 s/event to 3.2 s/event)[1].

The branching method is efficient but its drawback is that due to high combinatorics its computational and memory requirements can grow with time and can saturate the computing system. In order to prevent too much growth of combinatorics, only a limited number of branches from the N nearest hits in the validation gate are created. If limiting the maximum number of branches to $N = 4$ no efficiency loss is observed. However, the calculation time decreases 4 times (from 3.2 s/event to 0.88 s/event).

The geometry navigation based on the ROOT geometry package is a very precise method, but not efficient in terms of calculation speed. First, because it uses the detailed Monte-Carlo detector geometry, consisting of 800000 nodes. Second, because this algorithm is rather general and not optimized for the CBM setup. Thus, a simplified detector geometry and an optimized geometry navigation algorithm were developed. The simplified geometry is created by converting the detailed Monte-Carlo detector geometry into a simple one, which only consists of planes perpendicular to the beam direction. The navigation in such a simplified geometry has been significantly optimized. Overall, this leads to a factor 4.4 in computation speed (from 0.88 s/event to 0.2 s/event) without major efficiency losses in tracking (from 94.9% to 94%).

Since all modern CPUs have more than 1 core, usually 2-4 cores, and in future even more. Multithreading capabilities for the track reconstruction algorithm have been explored. As first step, the multithreaded nearest neighbour tracking has been

[1] Here and later laptop with Intel Core2Duo P8400 processor was used for tests.

developed. For a two core CPU a speed up factor of 1.6 has been achieved (from 0.06 s/event to 0.04 s/event).

4. Performance of the Tracking Algorithms

In order to test the track reconstruction algorithms, central Au+Au collisions at 25 AGeV beam energy were simulated with UrQMD [7]. These events were used to estimate the background in which the interesting signal, i.e. electrons or muons from the primary vertex were embedded. In order to enhance statistics 5 μ^+ (e^+) and 5 μ^- (e^-) were embedded in each event at the primary vertex. Compared to the overall multiplicity of charged tracks in the acceptance (appr. 800) they do not distort the overall conditions. GEANT3 [8] was used for transport through the CBM detector setup. Hits were calculated from the MC information using semi-realistic detector response.

The above described tracking routines were then tested on this dataset. A TRD track was accepted if having at least 8 hits out of 12. The resulting momentum integrated efficiency value is 94.7% for all tracks, and 91.0% for electron, i.e. signal tracks. The efficiency for matching TOF hits to global tracks is 92% for all tracks and 86% for electron tracks. The additional efficiency losses for electrons are probably due to the more complicated energy loss of electrons in particular in the TRD. From all reconstructed tracks 6% are ghost tracks.

For the MUCH detector reconstructed tracks were required to pass through the whole iron absorber of 2.25 m length. This is the condition which would be used for J/ψ reconstruction. Efficiencies and computational times from the three described tracking methods are listed in Table 1. The track finding method employing branching has the highest efficiency, however is also the slowest method. Efficiencies for the weighting approach are only lower by 1% but require less time. The simplest and fastest method in which just the closest hit is attached shows only a drop of 2.2% compared to the branching method. The fraction of ghost tracks is low in all cases.

Table 1

MUCH track finding efficiency and succeding TOF hit matching efficiency. The three methods described in the text are compared. The time given is without the improvements added for the acceleration of routines.

	Nearest Neighbor	Branching	Weighting
MUCH tracking, all tracks	91,7%	93,8%	92,8%
MUCH tracking, muon tracks	92,7%	94,9%	94,0%
TOF matching, all tracks	97,1%	99,2%	97,0%
TOF matching, muon tracks	97,2%	99,2%	97,0%
Time, s/event	0,20	0,88	0,31

5. Summary

Efficient and fast track reconstruction algorithms are a key ingredient to the CBM experiment. Based on track reconstruction in the silicon tracking system inside a dipole magnet, three track following methods for different detector setups were implemented. All methods are based on track following, the use of a Kalman filter and the calculation of a validation region. However, the discussed methods differ in the way hits are attached to the track. All three methods show efficiencies above 90%. First investigations and developments in order to speed up the algorithms showed acceleration potential of at least a factor 20. Overall, these tracking routines show, that the physics performance anticipated by the CBM detectors can be met with the current layout.

References

1. *Collaboration C.* Compressed Baryonic Matter Experiment. Technical Status Report: Techrep / GSI Darmstadt. — 2005.
2. *Höhne C., Rami F., Staszel P.* The Compressed Baryonic Matter Experiment at FAIR // Nucl. Phys. News. — 2006. — Vol. 16, No 1. — Pp. 19–23.
3. *Kisel I.* Event Reconstruction in the CBM Experiment // Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment. — 2006. — Vol. 566. — Pp. 85–88.
4. *Fruhwirth R.* Application of Kalman Filtering to Track and Vertex Fitting // Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment. — 1987. — Vol. 262, No 2-3. — Pp. 444–450.
5. The ROOT team. The ROOT Users Guide. — 2009. — `ftp://root.cern.ch/root/doc/Users_Guide_5_24.pdf`.
6. *Lebedev A., Ososkov G.* LIT Track Propagation for CBM // CBM note. — 2008. — `http://www.gsi.de/documents/DOC-2008-Dec-182-1.pdf`.
7. *Bass S. A. et al.* Microscopic Models for Ultrarelativistic Heavy ion Collisions // Progress in Particle and Nuclear Physics. — 1998. — Vol. 41. — Pp. 225–369.
8. GEANT — Detector Description and Simulation Tool CERN Program Library Long Writeup. — W5013.

УДК 519.254; 539.1

Быстрый алгоритм глобальной реконструкции треков для эксперимента CBM на ускорителе FAIR

А. Лебедев[*], К. Хене[†], И. Кисель[†], Г. Ососков[*]

[*] *Лаборатория информационных технологий*
Объединённый институт ядерных исследований
ул. Жолио-Кюри д.6, Дубна, Московская область, 141980, Россия
[†] *Центр имени Гельмгольца по исследованию тяжелых ионов*
64291 Дармштадт, Германия

Реконструкция траекторий заряженных частиц — это важная и сложная задача в эксперименте CBM (Compressed Baryonic Matter) на будущем ускорителе FAIR в Дармштадте. В работе представлен текущий статус программного обеспечения по глобальной реконструкции треков в эксперименте CBM. Глобальная реконструкция треков основана на методах слежения по треку и фильтре Кальмана. Эффективность реконструкции треков для центральных столкновений золото–золото при энергии 25 ГэВ на нуклон, смоделированных с помощью UrQMD составляет 93–95%. В эксперименте CBM должны быть обработаны терабайты входных данных при высокой интенсивности соударений, следовательно, чрезвычайно важно разработать быстрые алгоритмы реконструкции треков. Исследованы возможности по ускорению алгоритмов. Значительно оптимизирована работа с памятью и комбинаторные вычисления. Использование многопоточности позволило ещё больше ускорить алгоритм. В целом достигнуто 20 кратное уменьшение времени просчёта.

Ключевые слова: физика высоких энергий, эксперимент CBM, восстановление треков, фильтр Кальмана, фитирование треков.