Status of tracking in the TRD and MUCH detectors of the CBM experiment

A. Lebedev[1,2]*, C. Höhne*[3]*, I. Kisel*[1]*, and G. Ososkov*[2]

[1]GSI, Darmstadt, Germany; [2]LIT JINR, Dubna, Russia; [3]Justus Liebig University, Giessen, Germany

In this report the status of the track reconstruction in the Transition Radiation Detector (TRD) and muon system (MUCH) of the CBM experiment is presented. Global track reconstruction in the electron and muon setup of the CBM detector, i.e. with either TRD or MUCH system, is based on track following using reconstructed tracks in the STS as seeds. In the STS the track reconstruction is based on the cellular automaton method and provides initial track parameters as starting point for the following track prolongation. This ray tracing is based on the standard Kalman filter technique and is used for the trajectory recognition and estimation of track parameters in the TRD and MUCH. Tracks are prolonged subsequently from one detector station to the next adding hits in the detector stations.

The detailed layout of the detectors is still under investigation. In the high track density region of the TRD and MUCH stations, a pad layout is foreseen based on MWPC or GEM technology. For the downstream detector stations in MUCH, where track densities are low, straw tube chambers are under discussion. The first TRD station can be used as tracking station after the last MUCH absorber.

In the performed studies the algorithms were tested using central Au+Au collisions at 25 AGeV beam energy from UrQMD. In addition, for the reconstruction in TRD, 5 primary e^+ and 5 primary e^- with momenta 1 GeV/c $\leq$ p $\leq$ 10 GeV/c were embedded in each event. The performance inside the muon system was evaluated by embedding 5 primary μ^+ and 5 primary μ^- per event with momenta 2.5 GeV/c $\leq$ p $\leq$ 25 GeV/c.

Table 1: Track finding efficiency for muon tracks for five different MUCH geometries in %. NN – nearest neighbour algorithm; branch – branching algorithm

	MUCH geometry				
	MUCH1	MUCH2	MUCH3	MUCH4	MUCH5
NN	93.8	94.4	93.8	92.8	92.8
Branch	94.0	94.7	94.1	94.9	94.9

The tracking efficienciy for five different MUCH layouts was compared: 1) MUCH1 is the standard layout with 6×3 detectors including pad readout, 2) MUCH2 has 13 detectors with pad readout, 3) MUCH3 has 3×3 detectors with pad readout and straw tube detectors in the last 3×3 stations, 4) MUCH4 is the same as MUCH2 but with TRD after the last absorber, 5) MUCH5 is the same as MUCH3 but with the TRD after the last absorber, i.e. MUCH5 is currently the preferred layout. Table 1 lists the track finding efficiencies for these 5 MUCH geometries using two algorithms: the nearest neighbour as well as the branching. The reconstruction efficiencies differ slightly with respect to the MUCH layout option.

Concerning the TRD layout, the influence of the TRD detector position resolution on the tracking efficiency was studied. The position resolution for the x coordinate was fixed to 300 μm. In y direction it has been varied from 100 μm to 30 cm. Nearest neighbour tracking was used in this study. The track finding efficiency is presented in Fig. 1. The efficiency stays acceptable up to a position resolution of several centimeters. For larger polar angles, where track densities are lower, the detector resolution can be up to 6–7 cm.

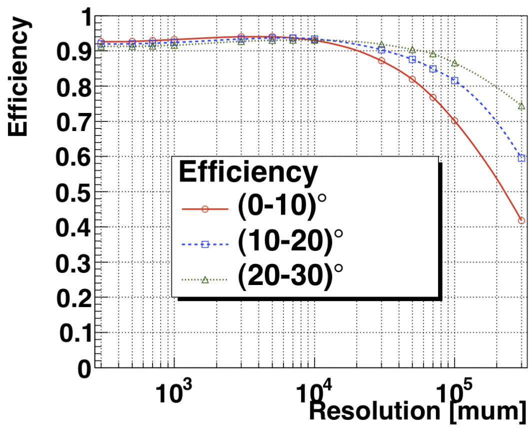

Figure 1: Track finding efficiency for TRD in dependence on position resolution for different polar angles

The speed of the tracking software is extremly important for data analysis in CBM. A fast parallel track reconstruction algorithm which uses available features of modern processors was further investigated. The algorithm uses two features of modern CPUs: a SIMD instruction set and multithreading. The results of the algorithm speedup for the MUCH system are presented in Table 2.

Table 2: Speedup of the track fitting (time is shown per track in μs) and track finding algorithm (time is shown per event in ms) for track reconstruction in MUCH

	Track fitting		Track finding	
	Time	Speedup	Time	Speedup
Initial	1200	-	730	-
Optimization	13	92	7.2	101
Vectorization	4.4	3	4.9	1.5
Multithreading	0.5	8.8	1.5	3.3
Final	0.5	2400	1.5	487