ACTS Tracking Performance in MPDRoot with Native TPC Geometry and Fast Clustering

S. Hnatič[1], *V. Krylov*, *A. Bychkov*, *O. Rogachevsky*, *J. Buša Jr*, *A. Moshkin*, *A. Krylov*

Joint Institute for Nuclear Research
6 Joliot-Curie, 141980 Dubna, Moscow Region, Russian Federation

MPD (NICA) is a world-class relativistic nuclear physics experiment in Russia, projected to run for a few decades and to collect heavy ion collisions data for physical analysis at the 100 PB scale.

The main challenge of the offline data processing software for the MPD experiment, the MPDRoot, is event reconstruction - the cascade of algorithms converting raw data signals into physical information, such as the momentum of charged particles, during the track reconstruction process. The significance of NICA future scientific discoveries depend directly on the quality of event reconstruction.

In this paper, we present the improvements obtained by implementation of native MPD TPC geometry into MPDRoot's ACTS TPC tracker and the replacement of the MPD default clustering by the new fast clustering and hit extraction algorithm. The new modules are inherently designed to be able to operate in the real experiment.

PACS: 07.05.Tp; 12.38.Mh

Introduction

The MPD detector at the NICA collider is scheduled for commissioning in 2026. While it is planned to obtain $\sim 10^{10}$ heavy ion collision events in the experiment, its physics performance was already extensively studied using simulated data [1, 2]. MPDRoot is the framework developed for the reconstruction of the simulated and experimental data from the MPD experiment [3]. Based on ROOT [4] and FairRoot [5], it uses stable and well-tested algorithmic suite for clustering & hit extraction and TPC track reconstruction in simulations [6].

The algorithms for the processing of the experimental data must adjust to real-life conditions, such as effects of noise, detector calibration [7] or detector misalignment [8]. Since the fast clustering & hit extraction algorithm from the online MPD Event Display [9] already has this feature, it was ported to MPDRoot. At the same time, the native MPD TPC tracking geometry was developed for the ACTS MPDRoot implementation [10], which is taking place in order to replace default MPDRoot's tracking algorithms by the state-of-the-art particle tracking technology [11]. This work includes the performance study of these two new modules.

[1]E-mail: hnatics@jinr.ru

ACTS in MPDRoot

Since the pilot implementation of ACTS in MPDRoot was written [10], the newest versions of ACTS library are regularly built and patches are written to adapt it to the specifics of the MPD software development and computing environment [12]. The present status of the MPDRoot's ACTS implementation is shown in Fig.1.

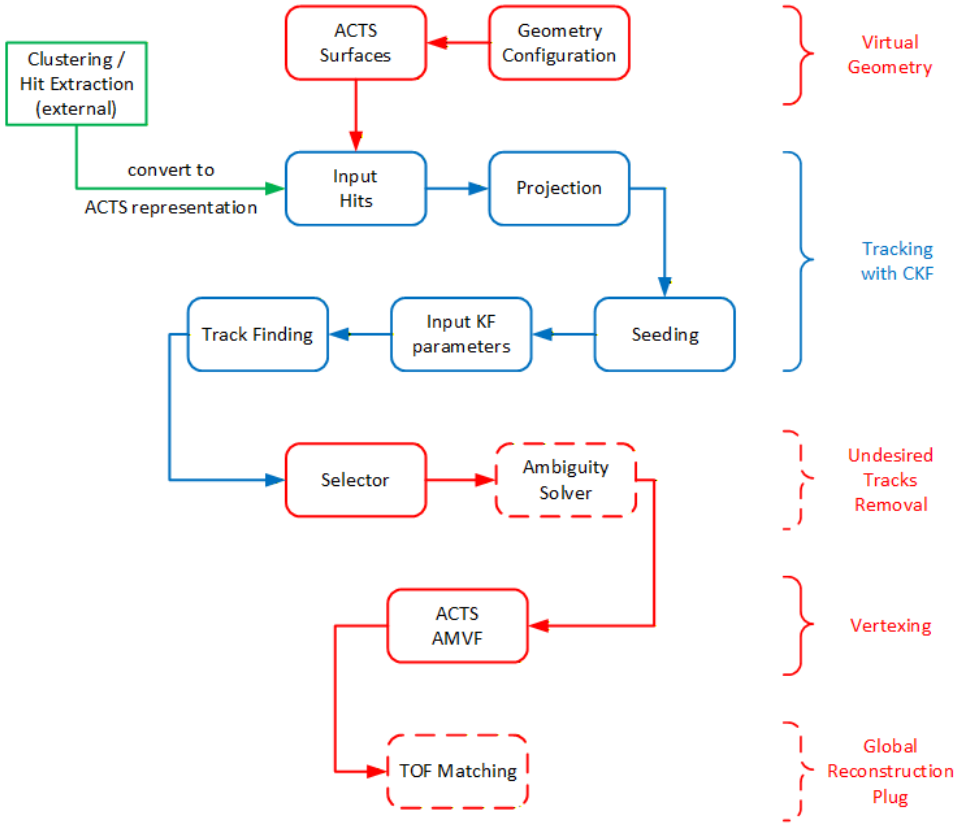


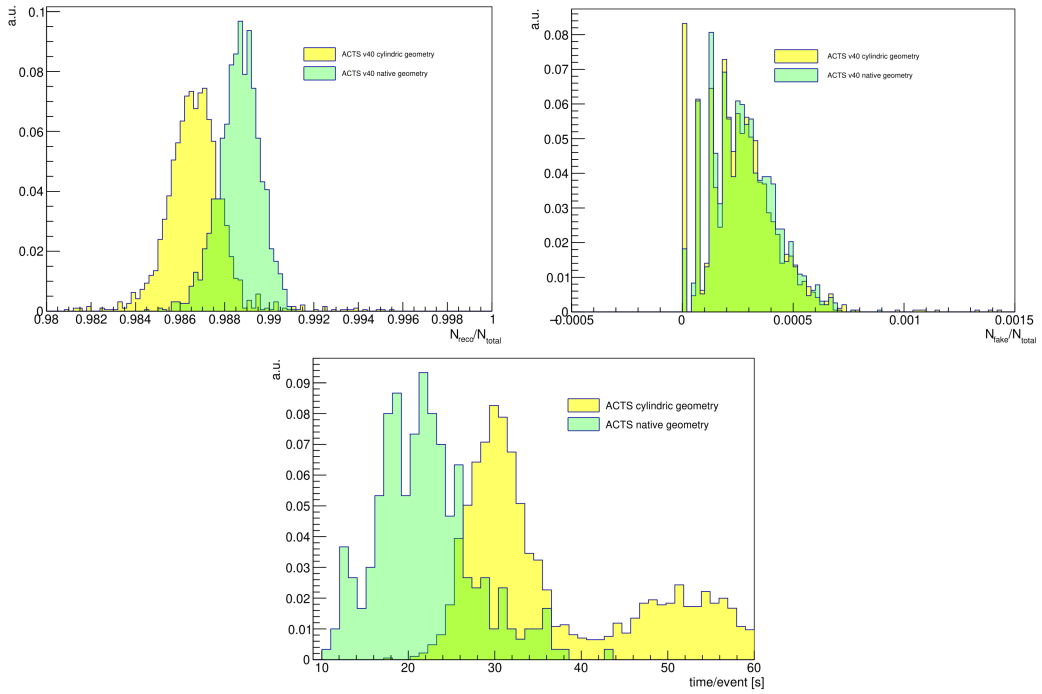
Fig. 1. ACTS pipeline in MPDRoot. Dashed modules are yet to be implemented.

Notice, that apart from the TOF Matching module, the Ambiguity Solver module is absent in calculations, as it was not implemented yet. This is not a hindrance for comparison of different modules against one another in terms of tracking performance [11]. It is important to note, however, that in the final version of the tracking procedure - after the Ambiguity Solver algorithm removes duplicates - the overall reconstruction efficiency will be slightly lower and the fake rate slightly higher [13, 14].

Cylindric vs Native ACTS TPC Tracking Geometry

In the initial implementation, TPC geometry was translated into ACTS compatible cylindric approximation with 180 sectors and 200 layers using

TGeometry plugin and TGeoDetector class from ACTS, analogically as done by sPHENIX collaboration [15]. This approach gave efficient tracking, but suffered from several weaknesses, the most detrimental being time consuming code maintenance, slowing down of calculations and significant increase in memory consumption. To resolve them, the native TPC tracking geometry (24 sectors and 53 padrows per sector) was developed in plain ACTS format to replace the cylindric approximation. Fig. 2 shows the tracking benchmark comparison of these geometries.



	Native Geometry	Cylindric Geometry
Efficiency (N_{reco}/N_{total} [%])	$\mu = 98.865$ $\sigma = 0.092$	$\mu = 98.667$ $\sigma = 0.133$
Fake ratio (N_{fake}/N_{total} [%])	$\mu = 0.027$ $\sigma = 0.014$	$\mu = 0.026$ $\sigma = 0.016$
Speed (time/event [s])	$\mu = 21.95$ $\sigma = 5.6$	$\mu = 36.73$ $\sigma = 10.27$

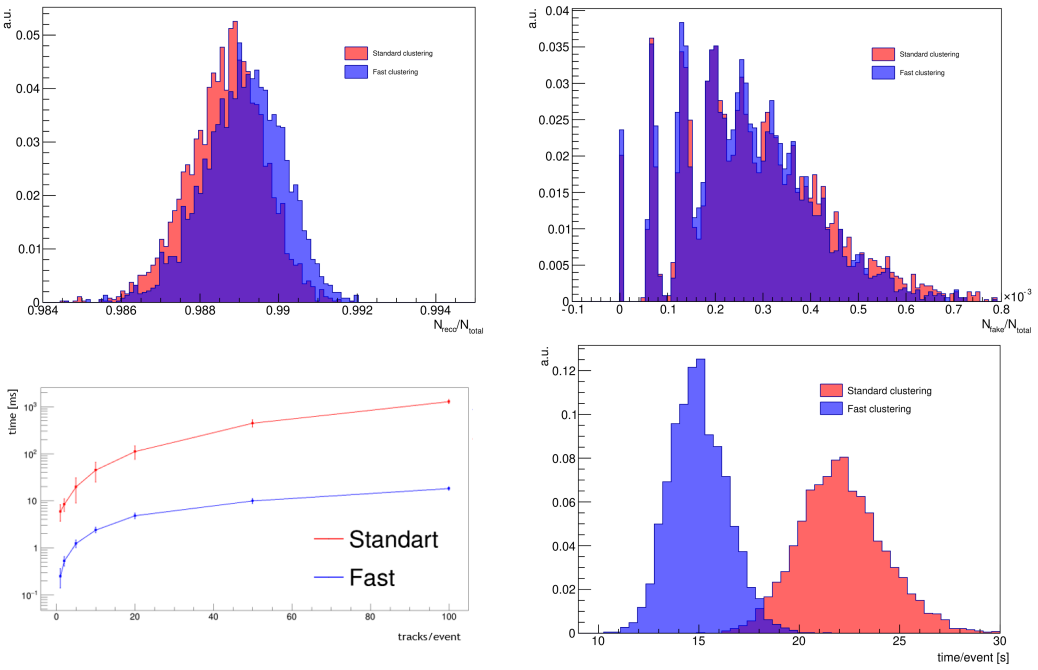
Fig. 2. ACTS performance with native TPC and cylindrically approximated tracking geometries.

The simulations were run on JINR's GOVORUN cluster [16], 200.000 events (2000 runs, 100 event per run), UrQMD generator, BiBi collisions at $\sqrt{s_{NN}} = 9.2\,\text{GeV}$, the tracks with at least 9 hits were considered. Tracking with native tracking geometry slightly outperforms cylindrical tracking geometry approximation in efficiency and results in significantly faster computational speed as expected.

Standard vs Fast Clustering & Hit Extraction

The fast clustering & hit extraction module was initially written for online visualization with focus on computational speed. It is experiment-ready, written in modern C++, designed to be maintainable, extendable and upgradable for the lifetime of the experiment [17]. It is multi-threaded (1 thread per TPC sector), has average event processing time $\sim$ 10ms[1] and is two orders of magnitude faster than the standard clustering.

Due to these advantages, the developments were made to increase accuracy of fast clustering & hit extraction algorithm to such level, so it can replace offline standard MPDRoot clustering. Fig. 3 shows the tracking benchmark comparison of both algorithms.



	Fast Clustering	Standard Clustering
Efficiency (N_{reco}/N_{total} [%])	$\mu = 98.910$ $\sigma = 0.102$	$\mu = 98.866$ $\sigma = 0.097$
Fake ratio (N_{fake}/N_{total} [%])	$\mu = 0.026$ $\sigma = 0.013$	$\mu = 0.028$ $\sigma = 0.014$
Speed (time/event [s])	$\mu = 15$ $\sigma = 1.42$	$\mu = 22.12$ $\sigma = 2.11$

Fig. 3. Performance comparison between standard and fast clustering & hit extraction algorithms. Upper row - tracking efficiency and fake ratio with ACTS tracker; bottom right - computational time per event without tracker; bottom left - computational time per event with ACTS tracker.

[1] $<$ 100ms for largest MPD experiment event multiplicities.

The simulations were run on LHEP DDC cluster, 300.000 events (3000 runs, 100 event per run), UrQMD generator, BiBi collisions at $\sqrt{s_{\mathrm{NN}}} = 9.2\,\mathrm{GeV}$, minimum hits per track: 9 for standard, 12 for fast algorithm[2]. Such combination of values of minimum hits per track was empirically found to be the most accurate, with fast algorithm being slightly disadvantaged against the standard clustering[3].

Conclusions

Two new modules were implemented into MPDRoot's ACTS TPC tracking: native MPD TPC geometry and the fast clustering & hit extraction algorithm. Tracking performance benchmarks against the previously used cylindric geometry and the standard MPD clustering were calculated. Native MPD TPC geometry and fast clustering & hit extraction algorithm improved tracking performance in key aspects: efficiency, fake rate and calculation speed. To have complete MPD TPC reconstruction, duplicate tracks must be handled by ambiguity resolution algorithm. Its implementation into MPDRoot's ACTS tracking will be the focus of our next work.

Funding

This work was supported by ongoing institutional funding. No additional grants to carry out or direct this particular research were obtained.

Conflict of interest

The authors of this work declare that they have no conflicts of interest.

REFERENCES

1. *Abgaryan V. et al.* [MPD Collaboration] Status and initial physics performance studies of the MPD experiment at NICA // Eur. Phys. J. A. — 2022. — V. 58., no. 7 — P. 140.

2. *Abdulin R. et al.* [MPD Collaboration] MPD physics performance studies in Bi+Bi collisions at $\sqrt{s_{\mathrm{NN}}} = 9.2$ GeV // Rev. Mex. Fís.,. — 2025. — V. 71., no. 4 — pp. 041201

3. *Rogachevsky 0., Bychkov A., Krylov V., Krylov A., Moshkin A., Voronyuk V.* Software Development and Computing for the MPD Experiment // Phys. Part. Nuclei — 2021. — V. 52, no. 4. — P. 817–820.

[2]Fast clustering & hit extraction algorithm generates about 25% more hits.

[3]0.15% more tracks to be reconstructed after running fast clustering & hit extraction.

4. *Brun R., Rademakers F.* ROOT — An Object Oriented Data Analysis Framework. // Nucl. Inst. & Meth. in Phys. Res. A.— 1997.— V. 389, Issues 1-2.— P. 81–86.

5. *Al-Turany M., Bertini D., Karabowicz R., Kresan D., Malzacher P., Stockmanns T., Uhlig F.* The FairRoot Framework. // J. Phys.: Conf. Ser.— 2012.— 219 042003

6. *Zinchenko A. et al.* Event Reconstruction and Physics Signal Selection in the MPD Experiment at NICA // Phys. Part. Nuclei.— 2021.— V. 52, no. 4.— P. 691–697.

7. *Bychkov A., Rogachevsky O, Hnatic S.* Implementation of Task for Calibration of MPD TPC Electron Drift Velocity // Phys. Part. Nuclei Letters.— 2024.— V. 21, no. 4.— P. 711–714.

8. *Kuzmin V.* MPD TPC Alignment // Physics— 2023.— V. 5, no. 2.— P. 508–516.

9. *Krylov A., Rogachevsky 0., Krylov V., Bychkov A., Voronyuk V., Moshkin A.* Web Interactive 3D Event Display for the MPD Experiment at the NICA Collider // Phys. Part. Nuclei— 2021.— V. 52, no. 4.— P. 821–825.

10. *Hnatič S., Buša J. Jr., Bychkov A., Krylov A., Krylov V., Moshkin A., Rogachevsky O.* Implementation of ACTS into MPDRoot // Phys. Part. Nuclei— 2025.— V. 56, no. 6.— P. 1471–1475.

11. *Ai X. et al.* A Common Tracking Software Project // Computing and Software for Big Science.— 2022.— V. 6.

12. *Buša J. Jr, Hnatič S., Korenkov V., Rogachevsky O., Vaľa M., Vrláková J.* Unified Software Development and Analysis Environment for MPD Experiment at NICA Collider // Modern Information Technologies and IT-Education— 2022.— V. 18, no. 1.— P. 176–182.

13. *Ai X., Huang X., Li T., Liu Y.* Performance of Track Reconstruction at STCF Using ACTS // EPJ Web of Conf..— 2024.— V. 295. 03029

14. *Allaire C., Bouvet F., Grasland H., Rousseau D.* Ranking-based neural network for ambiguity resolution in ACTS // EPJ Web of Conf..— 2024.— V. 295. 03022

15. *Osborn J.D. et al.* Implementation of ACTS into sPHENIX Track Reconstruction // Computing and Software for Big Science.— 2021.— V. 5, no. 23.

16. *Kutovskiy, N. and et al* Integration of Distributed Heterogeneous Computing Resources for the MPD Experiment with DIRAC Interware // Phys. Part. Nuclei— 2021.— V. 52, no. 4.— P. 835–841.

17. *Tockey S.* How to Engineer Software: A Model-Based Approach. // Wiley — 2019. — ISBN: 978-1-1195-4662-7.