PARALLELIZATION METHOD TO SPEED UP THE TRACK RECONSTRUCTION PROGRAM IN THE SPD NICA EXPERIMENT* **

G.A. Amirkhanova, M.Y. Mansurova

Department of Artificial Intelligence & Big Data
Al-Farabi Kazakh National University
71 al-Farabi Ave., Almaty, 050040, Kazakhstan

N.T. Burtebayev, G.A. Ososkov

Institute of Nuclear Physics, Ibragimova str.1, Almaty, 050032, Kazakhstan

A.S. Shomanov

Nazarbayev University, Kabanbay batyr str. 53, Nur-Sultan, Kazakhstan

Received 14 November 2022, accepted 15 November 2022,
published online 26 January 2023

Reconstruction of trajectories of charged particles (tracking) is one of the actual problems in experiments in high-energy physics. In the tracking program developed by a team of authors of the Al-Farabi Kazakh National University and JINR to process data registered by detectors located in the magnetic field of the experimental setup SPD planned in a complex of the JINR NICA collider, an algorithm is proposed for sifting out false tracks that arise during neural network tracking. This algorithm is based on a threshold criterion that calculates the quality of the helical line fit to the samples that make up the candidate track recognized by the neural network. In this paper, that continues this research, a method is proposed to significantly speed up the algorithm to weed out false tracks by paralleling it. The results of a comparative analysis of the computational speedup when paralleling them with the condition of preserving the efficiency of track reconstruction are shown.

DOI:10.5506/APhysPolBSupp.16.2-A12

* Presented at the IV International Scientific Forum *Nuclear Science and Technologies*, Almaty, Kazakhstan, 26–30 September, 2022.

** The work was supported by Program No. BR10965191 (Complex Research in Nuclear and Radiation Physics, High Energy Physics and Cosmology for the Development of Competitive Technologies) of the Ministry of Education and Science of the Republic of Kazakhstan.

1. Introduction

One of the current problems in high-energy physics experiments is the recognition of charged particle tracks. To obtain undistorted physical results, it is necessary to weed out false candidate tracks appearing in the tracking process. This work is a continuation of [1, 2] where a criterion for sifting out false tracks is the mean square error of fitting the helical line to the counts, of which the found candidate track consists.

The process of particle tracking plays an important role in high-energy physics data processing. To track data recorded by detectors located in the magnetic field of experimental setup SPD (Spin Physics Detector) at NICA (Nuclotron-based Ion Collider facility) collider under construction at the Joint Institute for Nuclear Research (JINR) [3], program TrackNetv3, based on application of deep recurrent neural network [4] is used. At this stage, the neural network is trained and tested on a training sample obtained by the Monte Carlo simulation of events for simplified detector geometry. However, during the neural network reconstruction of tracks, the appearance of false track candidates, formed from pieces of close neighboring tracks, noise samples, *etc.*, is inevitable. To sift them, it is required to develop special filtering algorithms responding to disturbance of particle trajectory smoothness, presence of kinks, outliers of separate measurements, *etc.*

When solving the problem of particle reconstruction, we faced the problem of limitations related to the computational power of the computer on which the calculations were carried out. To overcome these limitations, we used paralleling techniques.

2. Parallelisation method to significantly speed up the algorithm to weed out false tracks

In this work, we present 2 distinct parallel algorithms based on 2 different track fitting approaches. Firstly, the first method that we describe is based on fitting tracks by finding approximations to the helical trajectory of points in a 3D plane. This method relies on the circle fitting method that tries to find optimal parameters of the radius and coordinates of the center [5]. Secondly, we present a different fitting method which relies on polynomial fitting procedures [6]. This method has been found to approximate trajectories of the tracks with nearly the same accuracy, and here we present a comparison of both methods and introduce parallel algorithms to test their efficiency.

The parallel scheme, described in Fig. 1 applies for both of track fitting algorithms. It splits the task into processing of the events by multiple simultaneous processes in round-robin way. In the dedicated pool of processes, once the process finished computations for one event it can start working on the next one. Hence, in these algorithms we parallelize track fitting with making a parallel threads running in parallel and hence achieving simultaneous processing of multiple events at the same time. The authors of [7, 8] present methods that have a different approach — executing kernels for several events simultaneously. The HLT GPU framework allows to run independent processing components, each of which performs track reconstruction on the same GPU if there is enough GPU memory for all of them [9]. This approach can also load the GPU well but multiplies the memory requirement by the number of simultaneous queues.

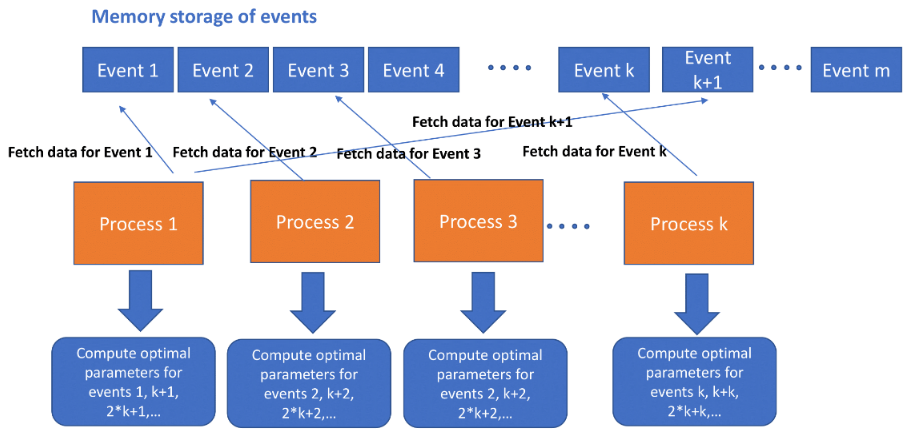

Fig. 1. Schematic diagram of the parallel algorithm.

In the proposed algorithm for computing optimal helix-loop parameters in parallel, we used the approach based on running parallel threads, where each thread performs a fitting procedure for events assigned to the threads according to the round-robin enumeration and circular loop ordering method. A schematic diagram of the parallel algorithm is shown in Fig. 1.

The table of all tracks contains the x, y, z coordinates of the event vertex, the identity number of the event N, the station number s, the detector number d, and the track number T within the event. Due to the large number of tracks formed by different convents, the idea of improving the performance of the process of determining the optimum helical parameters for each of the tracks arises. Thus, in our implementation, we propose a partitioning algorithm based on computation using multi-threaded computation. The partitioning algorithm is based on the round–robin approach based on the idea of sequential execution of tasks. If we have k threads, tasks are distributed in such a way that the first thread will compute the first track in the given table, the second thread will compute the second element, and so on, until we reach the k^{th} element (k^{th} thread is used to compute), after which $k+1^{\text{th}}$ element will be processed by the first thread again, and so on, until all the elements of the table are processed. The running time of the first parallel algorithm as a function of the number of threads is shown in Fig. 2.

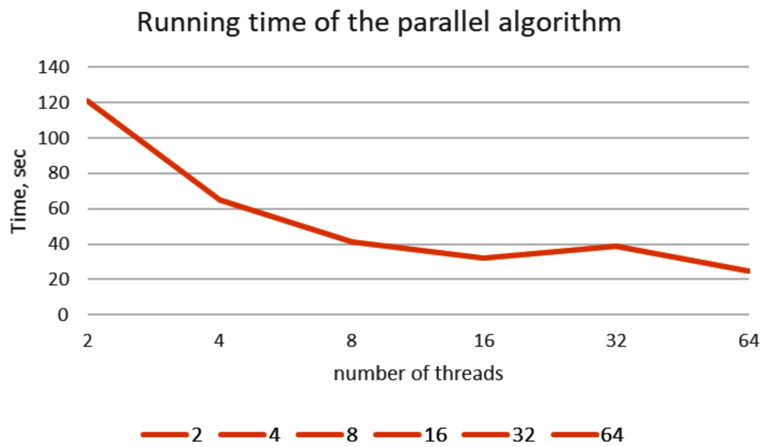

Fig. 2. Running time of the parallel algorithm as a function of the number of threads.

The input parameters for the second algorithm are the same as for the first algorithm and contain the coordinates of the event vertex x, y, z, event identification number N, station number s, detector number d, track number T within the event. The algorithm itself works on the basis of fitting polynomials of the third degree for track coordinates. Due to the more efficient computing cycle, this algorithm is characterized by a fast calculation procedure and, accordingly, this algorithm is less expensive for computing resources. Since the algorithm is iterative, it is required to determine the optimal number of iterations for convergence to the required computational accuracy. To do this, we applied methods for assessing convergence according to the formula

$$\frac{\log\left(\frac{\varepsilon}{|b-a|}\right)}{\log\left(R\right)}. \tag{1}$$

The algorithm is based on finding a third-degree spatial polynomial that approximates the coordinates of the counts belonging to the given track candidate. To find new values of the points at each iteration, locally minimizing the approximation error, the golden section search method is used. In this case, it turns out to estimate the number of search iterations according to the above formula (1). Searching by the golden ratio is an effective way to gradually reduce the minimum search interval. The key is to ensure that no matter how many points are evaluated, the minimum is within the interval defined by the two points adjacent to the point with the lowest value evaluated. Next, polynomials are constructed using standard square error minimization methods

$$E = \sum_{j=1}^{k} \left| p(x_j) - y_j \right|^2 . \qquad (2)$$

In reality, due to the inhomogeneity of the magnetic field of the SPD setup and the influence of such various factors that distort the particle trajectory, such as the Coulomb scattering and others, the helix in space ceases to adequately describe the trajectory. In addition, the iterative non-linear fitting method described above turns out to be slower than the linear approach and is more difficult to parallelize. Comparing both algorithms, we can highlight the main aspects of each method.

Note that after the introduction of parallelism, the number of iterations is no longer the main criterion. Thus, according to the results from Table 1, Algorithm 2 seems to be more optimal from the point of view of the main efficiency criteria mentioned above.

Table 1. Parallel algorithms in terms of the number of iterations, memory, and complexity.

	Number of iterations	Computational complexity	Memory usage
Algorithm 1	Low	High	High
Algorithm 2	High	Average	Average

Parallelisation is based on the multiprocessing library in the Python programming language. As an implementation, an algorithm for splitting the array of events into threads was used. The running time of the parallel algorithm depending on the number of threads is shown in Fig. 3.

The algorithm was tested on a multi-core compute node with the following characteristics: number of cores — 32, memory — 64 Gb, processor type — AMD Ryzen, disk memory (SSD) — 2 Tb.

This system supports up to 64 parallel threads, so in our experiment, we have shown computation time results for sequential code without parallelization, with 2 parallel threads, 4 parallel threads, 8 parallel threads, 16 parallel threads, 32 parallel threads, and 64 parallel threads.

In our experiment we used a track table consisting of 42102 tracks. Thus, the average execution time of a subroutine to calculate the optimum helix parameters for one track from the whole set is approximately 0.5×10^{-3} sec.

As we can see from the results, the algorithm shows a good speed-up, with a six-fold speed-up achieved. It is worth noting, however, that there is some deceleration of acceleration due to the additional memory fill factor.

Fig. 3. Comparative running time of two proposed parallel algorithms.

3. Conclusion

This article proposes a method for significantly speeding up the algorithm for filtering out false tracks by parallelizing it. The results of a comparative analysis of the acceleration of calculations during their parallelization with the condition of maintaining the efficiency of track reconstruction are shown.

REFERENCES

[1] P.V. Goncharov *et al.*, «Proceedings of the All-Russian Conference with International Participation: Information and Telecommunication Technologies and Mathematical Modelling of High-tech Systems», Moscow 2022, pp. 335–341.

[2] Zh. Zhunusova *et al.*, «Collection of Abstracts of the III International Scientific Forum "Nuclear science and technology"», Almaty, Kazakhstan 2021, p. 72.

[3] V.M. Abazov *et al.*, arXiv:2102.00442 [hep-ex].

[4] D. Baranov *et al.*, «CEUR Workshop Proc.», 2019, pp. 130–134, .

[5] N.I. Chernov, G.A. Ososkov, *Comput. Phys. Commun.* **33**, 329 (1984).

[6] K. Mørken T. Lyche, «Spline Methods», Centre of Mathematics for Applications, University of Oslo, 2008.

[7] A.S. Shomanov, M.E. Mansurova, *J. Phys.: Conf. Ser.* **1727**, 012018 (2021).

[8] M. Mansurova A. Shomanov, D. Akhmed-Zaki, «Parallel Computing Technologies, Lecture Notes in Computer Science, 10421», *Cham: Springer*, 2017, pp. 342–350.

[9] ALICE Collaboration (D. Rohr, S. Gorbunov, V. Lindenstruth), *J. Phys.: Conf. Ser.* **898**, 032030 (2017).