91-507

сообщения
объединенного
института
ядерных
исследований
дубна

E10-91-507

A.A.Glazov, I.V.Kisel, E.V.Konotopskaja,
G.A.Ososkov

TRACK FILTER ON THE BASIS OF A CELLULAR AUTOMATON

1991

1 Introduction

The experience of tens of investigations in experimental data handling in high energy physics ranged from bubble chambers with film data taking to modern electronic experiments (see for example surveys [1, 2, 3]) shows that the search for particle tracks on the projection of detected events is an inevitable part of any data processing procedure.

In many experiments in which the information is stored event by event to a permanent storage medium (the magnetic tape), the speed of the data acquisition ($10-100$ sec^{-1}) is higher than the speed of writing to the tape ($1-10$ sec^{-1}). And if these speeds are even comparable, there is the necessity to decrease the number of data recorded on the tape due to economy of magnetic tapes and computing time for off-line processing. This problem is especially important in the case of rare or forbidden decays, when the majority of events appear due to the random superposition of background tracks.

Methods are well known which are based on the local following of every separate track such as the variable slope histogram method [1] or the stringing method [2]. Unfortunately, in their practical applications in particular in the most actual cases of multitrack events these methods are too time consuming. Because of that 20 years ago an idea was born to split the data processing into two stages:

1. prefiltering, which allows a considerable decrease in the volume of the data to be written on the permanent storage due to the elimination of noisy counts and the extraction of the linear track elements (this stage could be implemented in the on-line mode by hardware realization of the filtering algorithms [4, 5, 6]);

2. the off-line stage of the track and event recognition (see for example [7]).

However, being oriented mainly to film data handling, these two-stage procedures appeared to be very expensive and not fast enough for very intensive streams of events in modern electronic experiments (up to 10^7-10^8 events per second). At the same time the significantly smaller noise level and the higher reliability of such track detectors as multiwire and strip chambers render it more possible to apply the principle component method [3, 8, 9] and in particular the global methods of track recognition. We mean here the methods when all tracks are at once searched on the view on the basis of different criteria of the closeness of these track points [10, 11, 12]. The main advantage of the global method is the possibility of parallelism [7].

It is necessary to point out here that the theoretical grounds of all above methods are based on two main assumptions: the independence and the normal distribution of measurement errors. But as noted in [13, 14, 15], this is not valid in the case of discrete detectors. The errors of measurements are in fact correlated and their distribution is not normal but uniform with the width specified by the wire or strip spacing. As shown in [14] without a proof, taking into account the discrete nature of the detector with the help of the Chebyshev metrics could improve the accuracy of track parameter reconstruction by a factor of $\sqrt{N}$, where N is the number of measurements. This evaluation was confirmed in [15] on the basis of the application of the algorithm, suggested by authors, to the simulated tracks. This algorithm was local in the sense of its sequential applying to every track from the given event. This prompts us to deal with looking for a global algorithm suitable for parallel implementing in the frame of the idea of two-stage processing: prefiltering in the first stage, which decreases significantly the information, and the event recognition in the second stage.

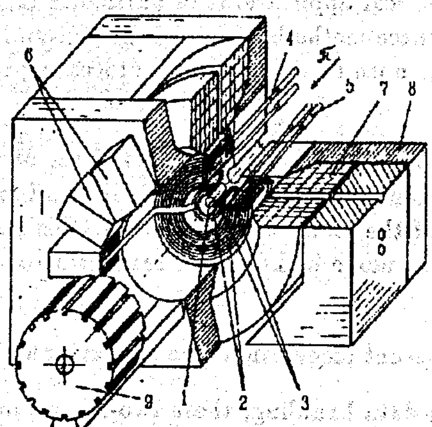

Figure 1: The ARES spectrometer: 1 - target; 2 - cylindrical proportional chambers; 3 - scintillators; 4 - lightguides; 5 - photomultipliers; 6 - magnet windings; 7 - magnet joke; 8 - poles of magnet; 9 - electronics.

The filtering method for tracks in discrete detectors based on the cellular automaton application is described below. Results of the successful application of this method to experimental data obtained on the spectrometer ARES (see fig. 1) during the search for the forbidden decay $\mu^+ \to e^+e^+e^-$ [20] and the study of the rare decay $\pi^+ \to e^+\nu_e e^+e^-$ [21] are given.

2 Physics background

Let us first consider some specific features of the discrete detectors. The setup for track detecting of charged particles is a system of several cylindrical multiwire proportional chambers and a target located in the center. A multiwire proportional chamber consists of a plane[1] grid with equidistant anode wires, which is placed between two parallel planes of grounded cathodes. Every wire is connected with its amplifier. The permanent positive potential is supported on it. When a charged particle crosses the grid, an electron avalanche appears in the chamber gas and causes the negative pulse on the nearest wire. Because the distance between the cathode planes is much less than the linear size of the chamber, the problem could be considered two-dimensional, i. e. we can neglect the bound effect.

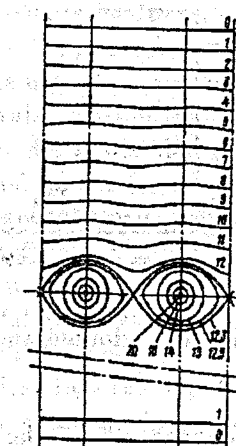

Figure 2: Equipotential lines in the multiwire proportional chamber.

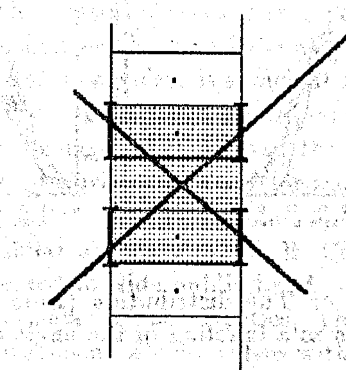

Figure 3: Replacement of a wire cluster by two vertical segments.

Let the wire grid in the proportional chamber be equidistant from the cathode planes. In practice the distance L between the anode and cathode planes ($\approx$ 6 mm) is larger than the step s between wires ($\approx$ 2 mm), the diameter d of

[1] The considerations below are also valid for cylindrical multiwire proportional chambers with a big radius, when the curvature effects are negligible.

wires is negligibly small (≈ 0.02 mm). In this case the electrostatic potential inside the chamber can be represented as [18]:

$$V \approx q\{(2\pi L/s) - \ln[4\sin^2(\pi x/s) + 4\sinh^2(\pi y/s)]\},$$

were $q = (V_0/2)[(\pi L/s) - \ln(\pi d/s)]^{-1}$ is the charge of the unit length of the wire; V_0 is the potential, applied to the chamber.

Equipotential lines in the proportional chamber are shown in fig. 2 [19]. One can see that the external equipotential lines are in fact parallel, at the same time they are concentric in the neighborhood of signal wires. If one takes into account that in the gas there are absorbing additions, then one can assume that there is some maximum distance h from the plane of signal wires beyond which we can neglect the electron gathering.

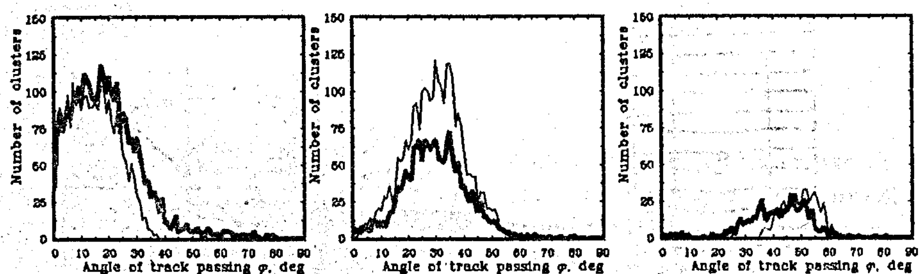

Figure 4: The distribution (from left to right) of single, double and triple clusters as a function of the angle of track passing.

It is confirmed by fig. 4. In this figure the dependence of the cluster width on the angle of the track pass through the wire chamber for single, double and triple clusters is shown[2]. Here the cluster is a continuous group of hit wires, and angle is counted from the vertical to the chamber. In the same figure the thin line is the cluster distribution for the model case, when the chamber is presented as a chain of rectangular cells with the maximal distance $h = 3$ mm of electron gathering. It is clear that this simple model is a satisfactorily good approximation for the description of cluster forming effects in the proportional chamber.

Thus every chamber can be represented as a chain of rectangular cells with the high degree of likelihood. Chambers are working according to the following

[2]Results are obtained on the ARES setup.

principle: if a charged particle track hits the cell, then the wire works inside it. If the track crosses few adjacent cells in the chamber, then they all work. One can note that if in the chamber few adjacent cells work, then the track crosses the left side of the lower cell and the right side of the upper one or vice versa (fig. 3). Otherwise it would either not have crossed all the hit cells or have touched one of missing ones. This is obvious for straight tracks and 'almost' true for curved ones with sufficiently small curvature. We neglect these possible errors. Then we replace each group of adjacent hits by only two segments (the corresponding sides of two utmost cells in this cluster).

3 Cellular automata

Cellular automata [16] are discrete dynamic systems whose behavior is completely determined by local mutual relations of states of these systems. The space is a uniform grid, every cell of which contains a few bits of information. Time is going by discrete steps. The evolution law is described by the only set of rules, let us say by a table. By this table every cell on each step can evaluate its new state according to the states of its neighbors. If the appropriate set of rules is given then this easy operational mechanism is sufficient for supporting the whole hierarchy of structures and events. Cellular automata give us useful modes for many fields in natural science. They created the general paradigm of parallel computations similar to Turing machines for sequential computations.

Quite an expressive example of the cellular automaton is J. H. Conway's beautiful game 'Life'. This game was presented to wide public by M. Gardner in *Scientific American* [17] and was very popular in the seventies. The game 'Life' describes the population of stylized organisms which evolves with a time due to antagonistic tendencies of birth and death. The individual of this population is the cell in state 1 and state 0 means the empty space. We can speak about living and dead cells. The measure of time going on is the change of the Life generations which obeys the following rules:

1. The cell neighbors are all living cells located in eight surrounding the giving cell.

2. If some cell has less than two neighbors it dies due to loneliness. If a cell has more than three neighbors it dies due to overpopulation.

3. If in the vicinity of an empty cell there are three living cells, then a new cell will born.

4. All births and deaths occur simultaneously in the moment of generation change. Thus the dying cell can help to bear the new one but the new born cell cannot resurrect the dying one. The death of one cell, decreasing the local population density, cannot prevent the death of another one.

Such a colony can grow all the time changing continuously its configuration, shape and the number of cells. More often, however, the colony becomes stable, cycling the same set of states. The length of the cycle is called the colony period. An example of the colony evolution is shown in fig. 5. On the left side the numbers of generations are denoted.

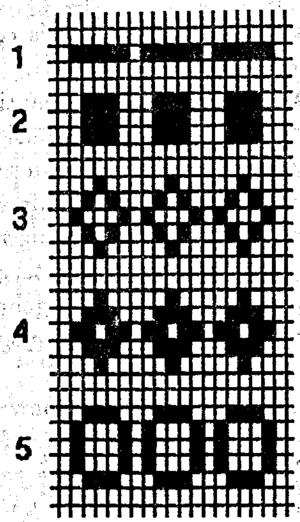

Figure 5: Example of 'Life' colony evolution.

Now, on the basis of this example, we can formulate the general rules for cellular automata:

1. The cell state is discrete (usually 0 and 1 although there can be automata with a larger number of states).

2. The number of neighbor cells is restricted, often they are the nearest cells.

3. The rules describing the dynamic of the evolution of the cellular automaton have usually a simple functional form and depend on the problem to be solved.

4. Cellular automaton system is clocked, i. e. the cell states change simultaneously.

4 Filter construction in the problem of track recognition

Let us try to create such a cellular automaton for event filtering. It has to sort out noisy hits leaving only experimental points which belong to tracks. The task of an automaton is:

- to leave cells belonging to track as living ones;
- to restore as newborns the gaps on the track caused by chamber inefficiency;
- to eliminate (to make dead) all cells which correspond to noisy counts;
- to divide the rest of living cells into groups, corresponding to different track candidates, on the basis of mutual neighborhood.

From there, on the ground of cellular automaton rules, we can formulate our requirements to the constructed automaton.

First, we define the living cell as the cluster, i. e. the continuous group of hit wires (see fig. 3). The dead cell is the empty one which does not contain any counts (remember, noisy clusters should die in process of the automaton evolution). To support the vital capacity of broken tracks (because of chamber malfunction) it is necessary to produce new cells-'phantoms' which should correspond to clusters in the case of correct chamber working. The phantom cells can also die and be reproduced again later but we have to follow them since only real clusters can appear in the final pattern. Thus in our case the cell has four states.

Secondly, we infer the rule for neighbor determination grounding on the discrete nature of our track detectors. As shown above the admissible track has to cross the opposite utmost segments of the cluster cells. The method of neighbor determination is build on this feature. There is the essential physics constraint: the track must leave the target and reach the last chamber. Therefore we can regard as the neighbors only those cells lying in adjacent chambers through which one can draw an admissible track. From there we determine the region of potential neighbors as the region which is swept up in adjacent chambers by all admissible tracks.

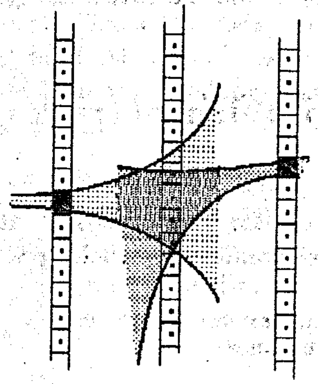

Figure 6: The birth condition of cell-phantom.

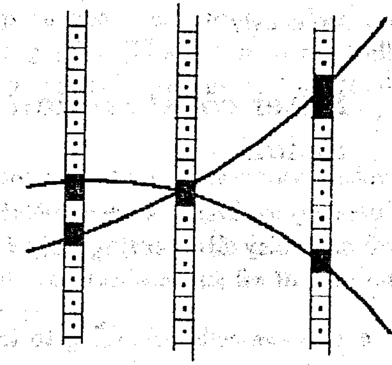

Figure 7: The crossing of two tracks.

Thirdly, let us consider the rules of automaton evolution in order to reject noise living tracks. Firstly, we have to restore missing clusters because of chamber malfunction. Due to the high efficiency of chambers (98 – 99 %) double malfunction of adjacent chambers is in fact impossible (the probability of this case is of order 10^{-4}) and we can neglect such cases. From here we can assume that if there are not neighbors in the region of potential neighbors, then it is a malfunction of the given chamber or electronics. In this case it is necessary to produce a phantom cell which is a neighbor of both the cells pointing to it (see fig. 6). Then we have to destroy noisy points, i. e. points having too few neighbors or too many of them. It is obvious the cell lying on the single track has exactly two neighbors. In the case of three-track events it is quite probable the two tracks are crossing. That means the cell on the track crossing has four neighbors (two from both sides). The probability of three tracks crossing in one point is negligible. Thus we have to destroy cells having less than two or more than four neighbors. The problem of utmost chambers is a separate one. To prevent dying out of tracks from both ends it is necessary to set the conditional chambers with numbers 0 and $(N + 1)$. They contain neighbors supporting all cells in real utmost chambers.

Fourthly, it is useful to separate the birth and death of cells into different stages. The order of this stages is essential. If we would eliminate noisy points in the first stage, then it could destroy the admissible tracks. Therefore it is reasonable to fulfil the birth in the first stage and the death in the second.

This order is repeated at every step of evolution to provide survival of broken tracks.

In order to sense the automaton arrival in the stable or cyclic state we suggest to calculate the checksum (CRC) for every generation and stop iterations in the case of checksum equality. In the method CRC (cyclic redundancy check) all bits of an array are considered coefficients of a binary polynomial and bytes of the checksum are the remainder from division of this polynomial to the known polynomial of degree 16.

5 Track filtering and results

The cellular automaton has been tested on three-track events observed in the experiment on the search for the forbidden decay $\mu^+ \to e^+e^+e^-$ [20] and the study of the rare decay $\pi^+ \to e^+\nu_e e^+e^-$ [21]. Only 10 chambers were used in this experiment. It means that every track contains on the average 10 experimental points (clusters).

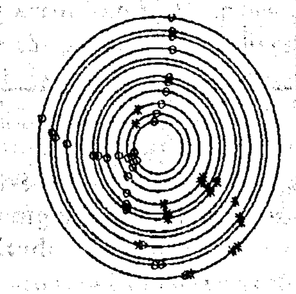

Figure 8: An example of how the cellular automaton works.

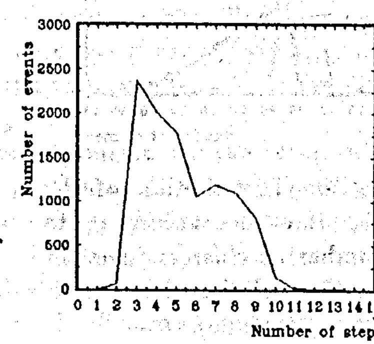

Figure 9: The number of filtering steps.

A result of the cellular automaton, which was created according to the above requirements, is shown in fig. 8. Crosses denote clusters eliminated by the automaton as noisy.

The advantages of the automaton are its simplicity and high speed of work (for example, one can replace the procedure of neighbor determination by looking up the table, which considerably accelerates the work). The execution time for one step is proportional to the cell number, one cell handling consists of looking up two tables. These tables contain the information:

1. about the presence of neighbors in adjacent chambers;
2. about the cell state at the next step according to the evolution rules.

In figure 9 the distribution of the number of filtering steps is shown. Let us point out that the automaton needs minimum two steps in order to make sure that its work is finished (checksums coincide). About five steps are needed on the average to finish the automaton work. The curve is decreasing slowly with the number of steps because of spiral tracks and random gathering of noisy points in real events (see the long 'tail' in fig. 10).

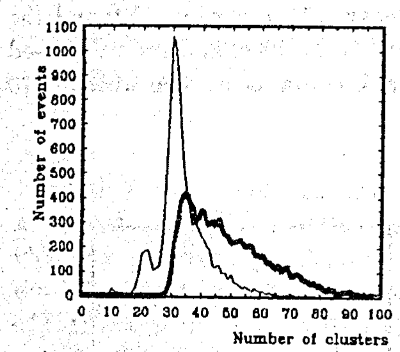

Figure 10: Distribution of the number of events according to the number of clusters in event before the cellular automaton and after (thin line).

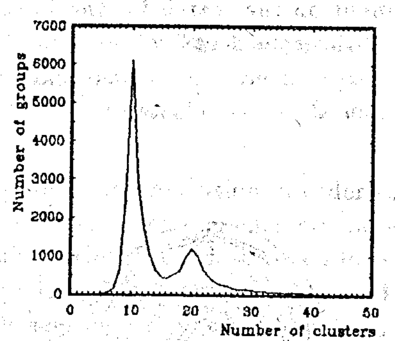

Figure 11: Distribution of the number of clusters in the group.

In figure 10 the distribution of the number of events according to the number of clusters in event is shown (the thin line shows distribution after the automaton work). In the original distribution one can see that due to the preselection there are no events with the number of clusters less than 30; also, the long noisy tail can be seen ranging up to 100 clusters per event. In the distribution obtained as a result of automaton working the picks are clearly seen in the region of one- and two-track events which were left in the previous stage of processing as well as three-track events (10, 20 and 30 clusters respectively). One can see the cellular automaton can reject the noise quite well. Now we can naturally determine the number of tracks according to the

number of clusters. Besides, the automaton can decrease the volume of data needed for further processing almost by a factor of 3: it rejects about 20 percent of events being non-three-tracks and decreases almost halves the number of experimental points from proportional chambers due to noise rejection.

The most important result for further processing is cells grouping according to the principle of possible belonging to track. In figure 11 there is the distribution of the number of clusters in the group marked by the automaton. One can see picks in the region of 10 clusters (one-track group) and 20 clusters (two-tracks group). It is obvious the local character of the cellular automaton (it takes into account the nearest neighbors only) does not allow it to separate close or crossing tracks. This problem will be solved in the next stage of processing.

6 Conclusion

The method of track filtering described above is the first application of cellular automata for track data handling in high energy physics, as we know. The results are quite successful: threefold reduction of input information with data grouping according to their belonging to separate tracks. The exciting results (even for the authors themselves) are remarkable because they were obtained with real data and lift up percentage of useful events, which simplifies and accelerates considerably their next recognition.

According to their nature the cellular automata are the ideal objects for making evolution algorithms parallel. In particular the minimal parallelism (at least one processor per chamber) increases the speed of calculation by the order of magnitude (there are other reserves). So the described cellular automaton for track filtering can be successfully applied in parallel computers (for example a transputer farm) and also in on-line mode if hardware implementation is used.

References

[1] G. A. Ososkov, Commun. JINR P10-83-187, Dubna, 1983.

[2] R. C. Strand, in: Digital Pattern Recognition, ed. by L. D. Harmon, Proceedings of the IEEE v. 60, No. 10, October 1972.

[3] H. Grote, Rep. Prog. Phys, 50 (1987) 473.

[4] A. J. Flavell e. a., in: Proceedings of the International Symposium on Data Handling of Bubble and Spark Chambers, JINR, D10-6142, Dubna, 1972, 248.

[5] P. Bacilieri, Nucl. Instr. and Meth. 135 (1976) 427.

[6] O. P. Fedotov, in: Proceedings of the Seminar on Physics Data Handling, Erevan, 1976, 198.

[7] G. Ososkov and M. Vajteršic, A fast parallel algorithm for the recognition of ionized particle tracks, Proc. of 6th Simpos. COMPSTAT'84, Physica - Verlag, Wien, 1984, p. 301.

[8] M. Hansroul e. a., in: Meeting on Programming and Mathematical Methods for Solving the Physical Problems, JINR D10-7010, Dubna, 1973, 460.

[9] G. A. Ososkov, A. C. Pospelov, Commun. JINR P10-91-444, Dubna, 1991.

[10] G. A. Ososkov, V. L. Pahomov, in: Meeting on Programming and Mathematical Methods for Solving the Physical Problems, JINR D10-11-11264, Dubna, 1978, 288.

[11] I. Bajla e. a., Computers and Artificial Intelligence, 3 (1984), No. 6, 527.

[12] C. I. Ioseliani e. a., Commun. JINR P10-86-666, Dubna, 1966.

[13] I. Duerdoth, Nucl. Instr. and Meth. 203 (1982) 291.

[14] F. James, Nucl. Instr. and Meth. 211 (1983) 145.

[15] N. I. Chernov e. a., Commun. JINR E10-91-361, Dubna, 1991.

[16] S. Wolfram (ed.), Theory and Applications of Cellular Automata, World Scientific (1986).

[17] M. Gardner, Scientific American, 223, 4 (1970) 120.

[18] G. A. Erskine, Nucl. Instr. and Meth. 105 (1972) 565.

[19] G. Charpak e. a., Nucl. Instr. and Meth. 62 (1968) 235.

[20] V. A. Baranov e. a., J. Phys. G: Nucl. Part. Phys. 17 (1991) 57.

[21] V. A. Baranov e. a., Jad. Fiz. (to be published).

Received by Publishing Department
on November 20, 1991.