An application of cellular automata and neural networks for event reconstruction in discrete detectors

I. Kisel, G. Ososkov

Joint Institute for Nuclear Research, Head Post Office, P.O. Box 79, 101000 Moscow, Russia

kisel@main1.jinr.dubna.su   ososkov@main1.jinr.dubna.su

We apply a cellular automaton to filter track data of charged particles registered in multiwire proportional chambers. A neural network is used not only to search for tracks but also to obtain their parameters. The fast algorithm based on the Chebyshev metrics is used for the vertex search. The results of testing the above methods with experimental data observed on the ARES spectrometer (JINR, Dubna) during a search for rare and forbidden decays are given.

Using the ARES spectrometer [1] as an example, let us consider in the XY projection (fig. 1) two specific features of a discrete detector like a multiwire proportional chamber (MWPC). First, the shape of external equipotential lines in a proportional chamber allows us to draw between wires imaginary equidistant planes perpendicular to a plane of wires. Second, due to electronegative additives in the chamber gas the distribution function of the electron collection is box-shaped, i. e. on both sides of the wire plane there exist two planes, so that all electrons that appeare inside them are collected by a wire. But any of electrons outside them are absorbed by the gas. Both these features allow us to consider the chamber to be a chain of imaginary rectangles, each of

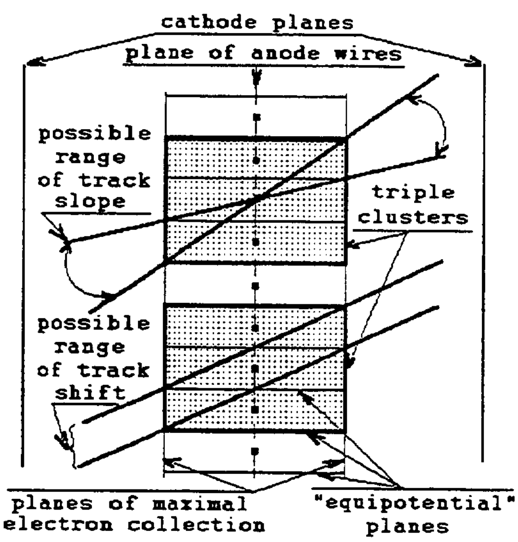

Figure 1: The model of MWPC functioning (wires are shown by bold dots).

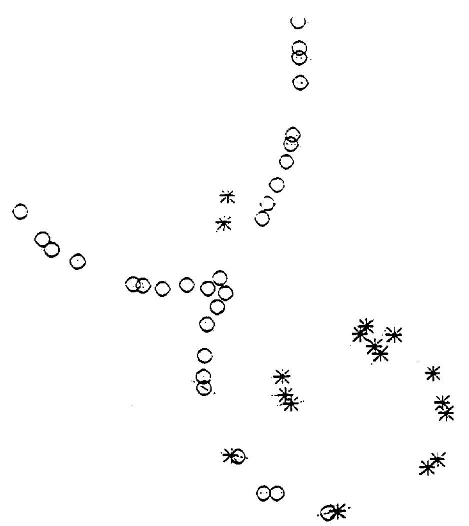

Figure 2: An example of the cellular automaton application.

them surrounding a signal wire. Two parallel sides of each rectangle are determined by the "equipotential" planes, two other sides are formed by the planes of maximal electron collection.

Whenever a charged particle track hits a rectangle, a wire inside it produces a signal. If a track crosses a few adjacent rectangles in the chamber, they all work, forming a cluster. In the latter case the track crosses the left side of the lower rectangle and the right side of the upper one or vice versa. The most important feature of our model of the discrete detector is the possibility of solving approximately the inverse problem of the track reconstruction (fig. 1). Knowing a cluster structure, one can make conclusion about a range of possible slopes of track passing through the cluster. Besides, when the slope is fixed, the cluster structure determines a possible range of points, where the track crosses the plane of signal wires. This model is fully confirmed by the results obtained on the ARES spectrometer [2].

The problem of data prefiltering in the analyzed experiments (the search for $\mu \to 3e$ and the study of $\pi \to 3e\nu$ decays) arose because of the high rate, the specific features of the beam duty cycle, chamber and electronics malfunctions, the presence of delta-electrons, returned tracks and other difficult-to-check reasons.

Using *terminology of the well-known game "Life"*, we can formulate some specific features of the automaton constructed for track filtering.

First, we define *cell* as a cluster.

Second, we suggest the rule for *determination of neighbors.* As was shown above, the cluster structure determines admissible angles of track passing. Thus only those cells can be treated as neighbors in adjacent chambers, through which one can draw a track according to the admissible angles. But there is an essential physics constraint in the analyzed experiments: every track must leave the target and reach the outer chamber. Therefore, we can regard as neighbors only those cells lying in adjacent chambers through which one can draw such an admissible track. From there we can also define *a region of potential neighbors* of a cluster as a region in the adjacent chambers that is swept up by all admissible tracks passing through the given cluster.

Third, let us consider *rules of automaton evolution.* To begin with, we have to restore clusters missed because of chamber malfunctions. In the usual case of the non-interrupted track, every cell has two neighbor cells in the adjacent chambers and lies on the overlapping of their regions of potential neighbors. If there is no cell in such an overlapping it is necessary to produce a cell that will be a neighbor of both cells pointing to it. Then we have to destroy noise clusters, i. e. cells having too few neighbors or too many of them. It is obvious that the cell lying on a single track has exactly two neighbors. In the case of two intersecting tracks the cell can have four neighbors. Thus we have to destroy those cells that have less than two or more than four neighbors. A special trick was done to prevent track dying off from utmost chambers.

Fourth, it is reasonable to fulfill *birth at the first stage* and *death at the second one* at every step of evolution to provide survival of interrupted tracks.

The final stage of the automaton evolution is attainment of a stable or cyclic state. To sense the automaton arrival in these states, we suggest to calculate a checksum (CRC) for every generation and to stop iteration by coincidence of checksums related the average track curvature in a point of each neuron allows us to use some robust procedure that significantly speeds up the NN convergence. Globality of E follows from its validity for all pairs lying along the track that is more suitable for the MFT-approach where, in principle, the long range interaction is required.

As a result of our neural network evolution, experimental points are grouped according to their relation to tracks and, even more, parameters of these tracks are found. The use of the cellular automaton at the previous step of data processing allows us to construct an initial configuration near the minimal energy. Therefore, the NN system evolves and converges on the average in 2–3 iterations, which means 0.3 sec per event with 30–35 coordinates for 3 MIPS computer. Track finding efficiency is higher than 98 %.

To complete event recognition in the XY plane one has to solve the vertex search problem. The Chebyshev metrics is very suitable for processing of the data from discrete detectors because errors of measurements are in fact correlated and their distribution is not normal but box-shaped with the width specified by the wire spacing [5]. Thus the vertex search problem is formulated as follows (parameters (a_i, b_i, R_i) determine the i-th track):

$$\mathcal{L}(x, y) = \max_i |\sqrt{(x - a_i)^2 + (y - b_i)^2} - R_i| \rightarrow \min \qquad (2)$$

with a condition $\sqrt{x^2 + y^2} \leq R_t$, where R_t is the radius of the target. As a consequence of this choice of the functional, search for the vertex (x, y) reduces to comparison of values of the functional in a few easy-to-find points. This method also provides a simple procedure of "pulling" the vertex to the target (see details in [6]).

Concluding, we would like to note that all three methods considered above were easily implemented as a complete software system for fast and reliable handling of real data in the XY projection. Our algorithms can obviously work with plane chambers and many-track events. Moreover, our NN approach can be also generalized to the case of an arbitrary magnetic field configuration and to $3D$-space geometry.

References

[1] V. Baranov et al., Nucl. Instr. and Meth. B17 (1986) 438.

[2] A. Glazov, I. Kisel, E. Konotopskaya and G. Ososkov, JINR Commun. E10-91-507, Dubna (1991) (submitted to NIM).

[3] A. Glazov, I. Kisel, E. Konotopskaya, V. Neskoromnyi and G. Ososkov, JINR Commun. E10-92-352, Dubna (1992).

[4] L. Gislén, C. Peterson and B. Söderberg, Neural Computation, 4 (1992) 727.

[5] N. Chernov, I. Kisel, S. Korenchenko and G. Ososkov, JINR Commun. E10-91-361, Dubna (1991) (to appear in CPC).

[6] A. Glazov, I. Kisel, E. Konotopskaya and S. Korenchenko, JINR Commun. E10-91-374, Dubna (1991) (to appear in CPC).