CERN-TH-6133/91

LAPP-TH-341/91

DF/IST-4.91

Cerenkov Ring Recognition Using a Non-Adaptable Network

T. Altherr

L.A.P.P., BP110,74941 Annecy-le-Vieux Cedex, France

and

J. Seixas

Theory Division, CERN, CH1211, Genève 23, Switzerland

and

Departamento de Física, IST Lisbon, Portugal

Abstract

We introduce a very simple and efficient technique to recognize one or several circles in a given pattern. The algorithm can (and in practical application should) be implemented as a massively parallel architecture, in which connections between units are not adaptable. We show examples of recognition of several (eventually overlapping) circles and a practical case of particle identification.

CERN-TH-6133/91

July 1991

1 - Introduction.

Adaptable Neural Networks (ANN) are known to give quite good answers to the problem of pattern recognition whenever the desired input/output connection is impossible (or very difficult) to obtain. This is the case in particular when we want to recognize a series of distinct patterns whose overlap is variable [1].

In some situations in High Energy Physics, however, we need only to recognize a *single* pattern scaled to different sizes. This is the case, *e.g.*, for the situation we are going to analyse in this paper, *viz.* Cerenkov circle recognition [2].

When a charged particle with a momentum p passes through a medium of index n such that $p > c/n$, it emits light around a cone whose angle θ is given by $\cos \theta = 1/\beta n$ where $\beta = v/c$ is the particle velocity relative to the speed of light. This is called the Cerenkov effect. From a front view, the cone appears as a circle. As the intensity of the Cerenkov light is usually low, only a few photons are emitted, which has the result that the Cerenkov ring appears not as a full ring but only as several dots lying on a circle. Hence, the challenge is to reconstruct the circle which passes through these points knowing that the circle can be deformed as the light originating from the particle usually goes through different materials and therefore can be diffracted or even reflected. Furthermore, one has to deal with some noise inherent to any low sensitivity detection.

As we will show here, a very simple and efficient method can be devised in which one makes no use of ANN. Instead one *can* use massive parallelism with a convenient architecture and obtain excellent results. Since ANN always pose the problems of high connectivity and other hardware implementation problems, the simplicity of the present approach may reveal itself quite useful and rewarding.

2 - Triangulation Method

To make a circle, three points is a minimum. But it is also the maximum number of arbitrary points for which there is a *unique* circle which contains the triangle formed by the three points. If we denote the coordinates of three arbitrary points by (x_i, y_i) , $i = 1, 3$, the circle which passes through these three points has for centre (x_0, y_0) with

$$\begin{aligned} x_0 &= \frac{1}{2} \frac{(x_2^2 - x_3^2 + y_2^2 - y_3^2)(y_1 - y_2) - (x_1^2 - x_2^2 + y_1^2 - y_2^2)(y_2 - y_3)}{(x_2 - x_3)(y_1 - y_2) - (x_1 - x_2)(y_2 - y_3)} \\ y_0 &= \frac{1}{2} \frac{(x_1^2 - x_2^2 + y_1^2 - y_2^2)(x_2 - x_3) - (x_2^2 - x_3^2 + y_2^2 - y_3^2)(x_1 - x_2)}{(x_2 - x_3)(y_1 - y_2) - (x_1 - x_2)(y_2 - y_3)}, \end{aligned} \quad (1)$$

and for radius

$$R = \sqrt{(x_i - x_0)^2 + (y_i - y_0)^2} \quad i = 1, 3. \quad (2)$$

Given a set of points, one can construct all possible triangles. This will give a set of centres and radii and if the initial set of points do form a circle, there will be only one centre and one radius. If there is some noise or if the circle is slightly deformed, there will be more than one centre and one radius, but if we count the number of triangulations which gives the same centre and same radius, the most likely circle will have the highest weight associated with its parameters.

Based on these simple considerations, we construct the algorithm as follows: in a first step, we perform the triangulations. The initial data (coordinates of the points) can be either analogic or digital. Let us consider that we have an input square layer of arbitrary size which has the initial data as active cells at the coordinates of the points. In most applications, it is clear that one does not need to recognize circles of any size, *i.e.*, circles which are too small (typically, these are the ones generated by the noise) or circles which are so big that they do not fit on the layer (but we can have eventually truncated circles—due to total reflection problems—whose centre can be near the edge of the layer). In order to fulfil these constraints, each cell is connected to all others, except for the nearest and the farthest ones. We have thus a high connectivity network, but the connectivity is not so high as in other networks like Hopfield or Back-Propagation. When the initial data is analogic, one just has to perform a do-loop on all the possible triangulations.

Then, for each triangulation, we project a weight of unit 1 onto a second layer (whose size is also arbitrary) at position (x_0, y_0) which is the centre of the circle, as well as the value of the radius. At the end of the triangulation, we obtain a second layer which has its most active cells on a spot centred around the centre of the circle. If we have several circles in the input layer, several spots will be formed in the second layer and the activity of the cells will grow accordingly to the number of points lying on each circles. It is clear that the precision on the exact location of the centre depends also on the granularity of this second layer.

At this stage we must have some kind of discrimination criterion to distinguish, among all the spots, the ones which are of interest. In the next sections we will show different types of discriminating algorithms.

Because we can discriminate the triangles which have one small angle, the triangles which are formed are mainly the ones of interest. Furthermore, the noisy points build triangles with the points lying on the circle but give very different centres and with a much lower activity. Indeed, if we have n points lying on a circle and only one noisy point, the activity of the central cell will be $C_n^3 = n(n-1)(n-2)/6$ while the activity

of the noisy cell will only be $n - 1$ and therefore smaller for $n \geq 4$. When the circle is deformed a bigger spot is formed on the second layer. However, although each cell of that spot can have a lower activity than a cell formed by a noisy point, the activity of the whole spot is much higher. One can also notice that our algorithm works well when only a part of the circle is present.

We believe that our approach of using this Non-Adaptable Network (NAN) which is specifically designed to recognize circles (although ellipse recognition would not require a major change of the strategy) is more efficient than using other neural networks like Hopfield or Back-Propagation. Indeed, the former would also be non-adaptable (since we would have to build a fixed cost function) but it is known to be quite slow. For the latter, one can imagine training a net to recognize circles but it might turn out that in most applications, the circles are too deformed or the noise is too important to render successful such an approach.

3 - General Circle Recognition

Let us first see an application of the method to a general case of circle recognition in a small net. We will consider a network whose input layer has 32×32 units and the circles are defined by randomly activating several units located along the rings.

As described previously, between the input and first processing layer, we perform all possible triangulations. In this processing layer the units corresponding to the position of the centres of possible circles are activated and their weights correspond to the number of times the unit has been chosen for a centre. Each unit carries also another activation value corresponding to the average of all circle radii with the centre located at this unit.* This first processing layer is a 64×64 unit layer in order to allow for circles only partly included in the input layer.

After this first processing stage we thus get a map of all possible circles present in the original pattern in the form of a set of spots of different sizes. To these spots we apply the spot reduction technique we have described in a previous publication [3] to get the most likely location and radius for each of the circles present in the initial pattern (see Fig. 1 for a sketch of the architecture).

Despite the simplicity of this approach the algorithm works quite well up to a number of three overlapping circles. In Fig. 2 we see the result in two cases with two circles.

*Notice that this restricts the applicability of the method since it does not allow the recognition of concentric circles.

Further refinement of the technique would of course be necessary in real applications. Here, however, we only want to show the power of the method even in quite complicated situations as these.

4 - Particle Identification

The main purpose of a RICH detector in High Energy Physics is to identify particles leaving identical signatures in calorimeters but giving different Cerenkov rings whose radius decreases as the particle mass increases.

With a toy Monte-Carlo we have simulated Cerenkov rings generated by pions, kaons and protons. We suppose that the geometry of the detector is such that the Cerenkov rings are always centred, and that an average of 10 photons is detected for each event. As the momentum of the particle can be measured independently, the three possible radii are known and one just has to test which radius corresponds to the detected photons.

This problem appears simpler than what we were doing in the preceding section. Here, we have only one circle whose centre is known. We again use the triangulation method but instead of making the complicated analysis for the second layer which has its cells activated according to the position of the circle centres, we do the following: first, we need only to take into account the centres which are near the origin. This allows us to eliminate a lot of centres which are formed by noisy points. Second, we add up the activity of all the cells in three classes corresponding to the three possible radii. The circle which is recognized is the one whose class gets the highest activity. With this method, we easily reach an average of 90% of correct particle identification. The program can be improved by imposing various cuts: a minimum distance is required between the points so as to eliminate clusters; also, one has to choose properly and according to the experimental setup the region around the origin which is projected onto the second layer. Finally, in the case of very low statistics (less than 5 points or so) the program checks precisely how many points lie on each circle. With these improvements, we have reached 97% recognition on a pion data set and 93% recognition on a kaon data set. Two examples are shown in Fig. 3.

The program has been written in FORTRAN and has been tested on a IBM 3090 and on a VAX 8000. On the slowest machine (VAX), it takes ~ 4 sec to analyze a sample of 100 events.

5 - Conclusion

We have developed a Non-Adaptable Network which is able to recognize Cerenkov rings. The algorithm is based on a triangulation method which defines a set of likely circles whose centres and radii are projected onto a layer which can be treated either by using a spot reduction technique or by other means according to the problem under study. In a simulated situation for a RICH detector, an average of 95% efficiency has been achieved in particle identification.

Note that the simplicity and – to a certain extent – low connectivity of the approach point to further general application of the method eventually on raw data coming from a RICH detector. This will, of course, only be possible to test when we use real experimental data, which we hope will be possible in the near future. On the other hand, since the program is fast and the connectivity is not overwhelmingly big, it might be possible to implement it in hardware for on-line analysis of detector data.

Copies of the programs are available from ALTHERR@FRCPN11.

Acknowledgements

We would like to thank S. Tzamarias, A. de Angelis and M. Dracos from DELPHI for illuminating discussions and remarks concerning the physics of the DELPHI RICH detector.

References

- [1] D. E. Rumelhart and J. L. McClelland, *Parallel Distributed Processing*, The MIT Press (1986).
- [2] T. Ekelof, CERN preprint CERN-PPE/91-23 (1991).
- [3] T. Altherr and J. Seixas in *New Computing Techniques in Physics Research*, Eds CNRS (1990).

Figure captions

Figure 1 : A schematic representation of the net architecture.

Figure 2 : Typical Cerenkov rings generated and recognized by our program. The dashed lines correspond to the generated circles and the solid lines to the recognized ones.

Figure 3 : Two typical events simulating a real situation in a RICH detector. The three possible circles corresponding to pion (dashed), proton (dotted) and kaon (dash-dotted) hypotheses are shown and the one which is recognized is in solid line.

Fig. 1

Fig. 2.a

Fig. 2.b

Generated: PION -- Recognized: PION

Fig. 3.a

Generated: KAON -- Recognized: KAON

Fig. 3.b