Possibilistic Rings Detection for RICH Pattern Recognition

Anna Maria Massone
IPHE - Institut de Physique des Hautes Énergies
Université de Lausanne
BSP, 1015 Lausanne
Switzerland
INFM - Istituto Nazionale per la Fisica della Materia
Via Dodecaneso, 33
I-16146 Genova, Italy
massone@ge.infm.it

Léonard Studer
IPHE - Institut de Physique des Hautes Énergies
Université de Lausanne
BSP, 1015 Lausanne
Switzerland
Leonard.Studer@iphe.unil.ch

Francesco Masulli
DISI - Dipartimento di Informatica e Scienze dell'Informazione
Università degli Studi di Genova
Via Dodecaneso, 35
I-16146 Genova, Italy
INFM - Istituto Nazionale per la Fisica della Materia
Via Dodecaneso, 33
I-16146 Genova, Italy
masulli@disi.unige.it

Abstract

In this paper we present some results obtained using an algorithm whose core is the Possibilistic C-Spherical Shell algorithm, to detect complex images full of rings without any preliminary knowledge of number and position of the rings. This algorithm depends on the choice of several critical parameters. The statistical study described in this paper lead to high success rates even for large number of rings.

Keywords: RICH counters, Possibilistic Clustering, Possibilistic C-Spherical Shell Algorithm.

1 Introduction

In 2005, the Large Hadron Collider (LHC) at CERN Geneva will be commissioned. Proton against proton will collide at an energy of 14 TeV. Four colliding points will be instrumented. One of the four experiments, LHCb[1], is dedicated to the study of CP violation in the B meson system. CP violation is an ingredient for the explanation of the matter dominance in our Universe. LHCb will use various detectors; among them two RICH counters (Ring Imaging CHerenkov) which will identify the type of stable charged particles (π, K, p, μ and e) produced in the decay of B mesons. If a charged particle goes through a dielectric material at a speed greater then the speed of light in this material, photons are emitted at a characteristic angle θ from the charged particle flypath. The Cherenkov relation is given by $cos\theta = 1/(\beta n)$ where β is the ratio of the speed of the particle to the speed of light in vacuum and n is the index of refraction of the medium. By a clever arrangement of focusing mirrors, it is possible to collect and detect the Cherenkov light emitted by the charged particle on a surface where this light forms rings. The ring diameter is a function of θ. Hence measuring the ring diameter allows measuring β and, assuming that another LHCb detector measures the momentum of the charged particle, its type can be derived. The pattern recognition problem in RICH counters is then to identify an unknown number of imperfect roughly-circular rings made of a low number of discrete points in presence of background points. We also assume that the centers of the rings are unknown. This last hypothesis is partially realistic as some of these centers could be provided by another LHCb detector with finite efficiency. Due to the constraints of the problem, distorted circles, partial information, background, randomness and a huge number of image to process, we turn to soft computing algorithms for their robustness, their tolerance to uncertain configurations and their potentially great processing speed. In this paper we describe our work in progress to address this pattern recognition problem. The proposed Possibilistic Rings Detector (PRD) algorithm is tested on artificial images made of a number of circles variable from 10 to 40 (25±5 points each) and a fixed level of background (1%). In a previous work [1] we estimated the intrinsic robustness of the algorithm in function of noise level (outliers rate) up to 100% (which is much higher than expected) and we showed that the algorithm is very tolerant and robust to noise level. Here the problem we face is the recognition of an unknown number of rings whose centers positions and radii are assumed to be unknown. The PRD automatically finds both the number of circles and their

[1] see: http//lhcb.cern.ch/

position with high precision and efficiency.

2 Possibilistic C-Spherical Shell Algorithm

The core of the PRD algorithm we propose is the Possibilistic C-Spherical Shell algorithm [2, 3] (PCSS) that is based on a possibilistic approach to Fuzzy Clustering methods. Moreover, in our case we had to extend the PCSS to solve some special problems. The PCSS algorithm searches for prototypes β_j consisting of the couple $(\bar{\mathbf{c}}_j, r_j)$, where $\bar{\mathbf{c}}_j$ is the center and r_j the radius of the j-th circle. Given a generic point $\bar{\mathbf{x}}_k$, belonging to a set of N samples, a distance measure from the prototype can be defined as:

$$d_{jk}^2 = (\|\bar{\mathbf{x}}_k - \bar{\mathbf{c}}_j\| - r_j)^2 \quad (1)$$

and the objective function iteratively minimized:

$$J(L,U) = \sum_{j=1}^{C}\sum_{k=1}^{N} u_{jk} d_{jk}^2 + \sum_{j=1}^{C} \eta_j \sum_{k=1}^{N} u_{jk}(\ln u_{jk} - 1) \quad (2)$$

where L is the set of C prototypes, $U = [u_{jk}]$ is the membership values matrix and η_j a parameter representing the zone of influence of a cluster. Update equations can be written for $(\bar{\mathbf{c}}_j, r_j)$ and u_{jk} in order to minimize the objective function [1].

In particular the update equations for u_{jk} are:

$$u_{jk} = \exp\left(-\frac{d_{jk}^2}{\eta_j}\right) \quad (3)$$

For the η_j parameters, we did not use the original definition given by authors [3], but we introduced an ad hoc definition for the case of spherical shells detection. We interpreted the distance of each point from the center of the clusters $(\|\bar{\mathbf{x}}_k - \bar{\mathbf{c}}_j\|)$ as function of a radius percentage p $(r_j \pm p r_j)$ and we calculated the η_j values for which the membership value of a point in a cluster becomes 0.5.

For these values:

$$u_{jk} = \frac{1}{2} = \exp\left(-\frac{p^2 r_j^2}{\eta_j}\right) \quad (4)$$

and so

$$\eta_j = \frac{p^2}{\ln 2} r_j \quad (5)$$

Tuning the p parameter, we can control the *fuzziness* of the clusters.

3 Cardinality and Collapse Criteria

Characteristic of the PRD algorithm is its capability to find the number of rings without any a priori knowledge. Anyway, an initial overestimated value (C_{ini}) is needed as starting point. In our previous work [1] we showed that using realistic databases consisting of at least 10 circles, the algorithm can not find the whole set of rings in a single iteration. The correct ones are less than the expected ones and a certain number of them are completely wrong. To distinguish between "good" and "bad" circles we introduced the so called *Cardinality Criterion*: noisy circles are generally less densely populated than the good ones and then we can introduce a threshold (THR_{CARD}) in order to automate the choice. If the cardinality of a circle is greater than the threshold the ring is accepted, otherwise it is rejected. Besides that, between accepted rings often the algorithm finds the same circle redundantly. A *Collapse Criterion* is introduced to identify coincident rings: two rings are identical if their distance in a three-dimensional space $(\bar{\mathbf{c}}_j, r_j)$ is less than another fixed threshold (THR_{COLL}). At this stage patterns belonging to "good" circles have very high membership values and can be removed from the database by introducing a threshold, the so called α-cut (α), in order to clean the database from sample points building unquestionable rings. In order to find the remaining rings it is necessary to reiterate the algorithm on the new "cleaned" database; the number of rings to be found now depends on the surviving patterns. At each step the algorithm can find a certain number of correct circles and at each step the database cardinality and the number of undetected rings are progressively reduced. The algorithm ends up when the remaining patterns are supposed outliers (the number of surviving samples is less than a fixed threshold THR_{DATA}) or when the performed iterations reach a maximal value $(STEP_{MAX})$. For the last iterations, a second threshold smaller than the previous one is needed for the Cardinality Criterion (THR_MIN_{CARD}). In fact at this point the "cleaned" database is just consisting of very imperfect circles with holes and arcs that are fragments of rings already recognized and removed. Note that our algorithm can infer perfectly valid circles from points forming partial circles (arcs).

4 Performances of the PRD Algorithm

The performances of the PRD algorithm depend on the choice of the cited critical parameters, whose values can be tuned. In Table 1 the whole set of parameters to be optimized and their meaning are resumed. We tested our algorithm on several synthetic databases consisting of 10, 20, 30 or 40 circles whose radii may have values between 0.5 and 1. Each circle is made of 25±5 samples and a level of 1% of added random noise is considered. For each set of critical parameters, we tested the algorithm 100 times. At each new trial a new database is generated completely random. No constraint is made on the position of the centers or on the local density of points. A few circles could be superimposed. In Figure 1, a database consisting of 40 rings is shown while in Figure 2 the whole set of recognized circles (after 11 iterations) is superimposed to the input database. Finally, in Figure 3 a zoom on a zone particularly full of rings is shown. In a typical test on 10 circles, 2-3 iterations of the algorithm are sufficient. 4-5 iterations are necessary on databases consisting of 20 circles while for the cases of 30 and 40 rings, the necessary steps increase to 5-10 and 10-15 respectively. It is worth noting that the effective iterations performed from the algorithm are more than the cited ones due to the fact that it could find no "good" rings (all the cardinalities under threshold) for a certain number of steps (case of circles densely superimposed or intersecting).

In order to study the dependency of the algorithm performances from the critical parameters, we considered 4 different sets (Table 2). For sets 1 and 2 we did not test the algorithm on databases consisting of 40 circles owing to the fact that performances was bad even in presence of 30. Besides that, the core of the algorithm (PCSS) performs a certain number of internal iterations until, for two successive iterations t and $t+1$ and for each prototype β_j:

$$d(\beta_j^{t+1}, \beta_j^t) < \varepsilon \qquad (6)$$

We first set $\varepsilon = 0.001$ as in Table 2, but we verified that in very few cases the PCSS did not reach convergence. Then we decided ε to change to 0.01 if the PCSS does not reach convergence in the first 100 iterations. In Table 3 the success percentages for the four sets of parameters are shown. In those cases the algorithm found exactly the 10, 20, 30 or 40 expected rings. As we can see, no appreciable difference is evident in presence of 10 circles, but increasing the number of circles to be found, the performances change considerably and the choice of the parameters becomes very important.

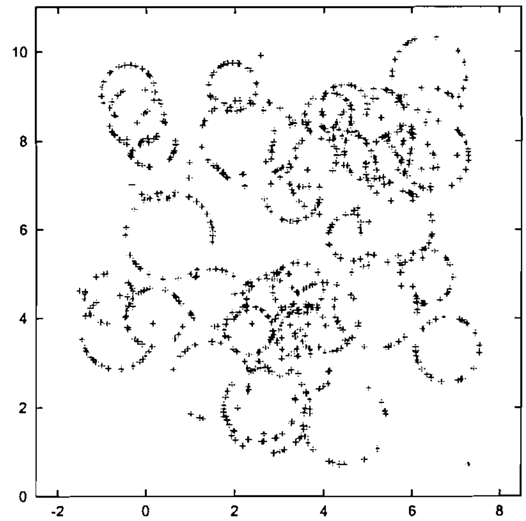

Figure 1: Database consisting of 40 circles.

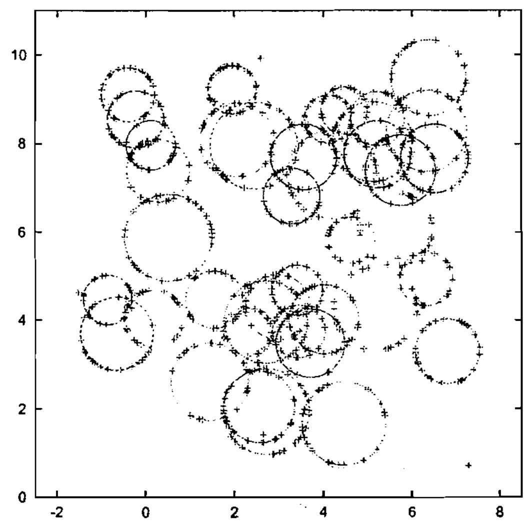

Figure 2: Final solution after 11 iterations.

5 Conclusion

We have presented results using the Possibilistic Rings Detector Algorithm, whose core is the Possibilistic C-Spherical Shell algorithm, to recognize

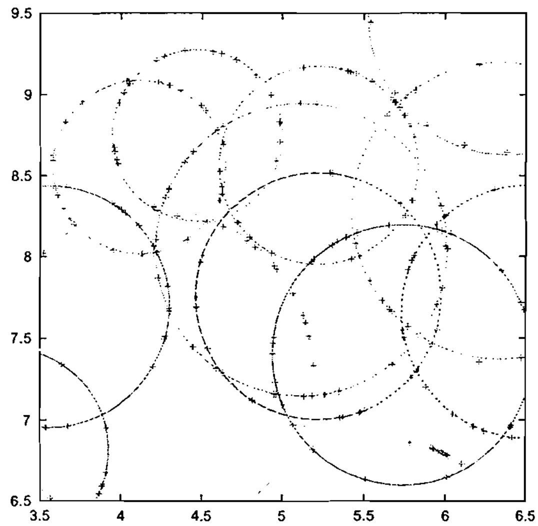

Figure 3: Zoom on a zone particularly full of rings.

Table 1: Critical Parameters.

Rings to be found	C
Initial circles	C_{ini}
Threshold for the cardinality	THR_{CARD}
Final thr. for the cardinality	THR_{CARD_min}
Threshold for the collapse	THR_{COLL}
Accuracy in the PCSS	ε
Threshold for the $\alpha-$cut	α
Influence region of prototypes	p
Minimum number of samples	THR_{DATA}
Maximum allowed iterations	$STEP_{MAX}$

rings on images without any preliminary knowledge of number and position of the rings. These rings are roughly circular, built with discrete points and noise randomly scattered in the image. The PRD algorithm has been shown very powerful in detecting complex images full of circles. Besides that, the rings are not requested to be complete, only arcs are sufficient to recognize the underlying rings. This automatically solves the realistic problem of half circles mentioned in the introduction while the fuzziness included in the algorithm adds tolerance to the imperfect circular rings. The PRD algorithm depends on the choice of several critical parameters, whose values can be easily tuned. A statistical study on the performances of the PRD algorithm tuning the set of parameters showed that it can be optimized and that the performances can be dramatically improved. Currently we are studying the application of the PRD algorithm to LHCb simulated RICH ring.

Table 2: Critical Parameters sets.

	Set 1	Set 2	Set 3	Set 4
C_{ini}	$3*C$	$3*C$	$3*C$	$3*C$
THR_{CARD}	15	15	15	18
THR_{CARD_min}	5	10	10	7
THR_{COLL}	0.001	0.001	0.001	0.001
ε	0.001	0.001	0.001	0.001
α	0.96	0.96	0.96	0.96
p	0.1	0.1	0.1	0.1
THR_{DATA}	10	10	10	10
$STEP_{MAX}$	15	15	30	50

Table 3: Success Percentages

C	**10**	**20**	**30**	**40**
Set 1	96%	71%	19%	
Set 2	99%	85%	54%	
Set 3	99%	92%	72%	43%
Set 4	99%	97%	88%	73%

Acknowledgements

This work was partially funded from the INFM Project *Global Search in Clustering.*

References

[1] A.M.Massone, L.Studer and F.Masulli (2000). Pattern Recognition in RICH Counters using the Possibilistic C-Spherical Shell Algorithm. In *Proceedings of the Fourth International Conference on Knowledge-Based Intelligent Engineering Systems & Allied Technologies KES'2000*, pages 792-795, Brighton, UK. IEEE 2000

[2] O.Nasraoui, R.Krishnapuram and H.Figui (1992). The fuzzy c-spherical shell algorithm: A new approach. *IEEE Transactions on Neural Networks*, 3(5):663–671.

[3] R.Krishnapuram and J.M. Keller (1993). A possibilistic approach to clustering. *IEEE Transactions on Fuzzy Systems*, 1(2):98–110.