Possibilistic Clustering Approach to Trackless Ring Pattern Recognition in RICH counters

A.M. Massone [a,b,*], L. Studer [b,*], F. Masulli [c]

[a] *CNR-INFM LAMIA, Via Dodecaneso 33, 16146 Genova, Italy*

[b] *IPHE - Institut de Physique des Hautes Energies, Université de Lausanne BSP, 1015 Lausanne, Switzerland*

[c] *Dipartimento di Informatica - Università di Pisa Largo B. Pontecorvo, 3 I-56127 Pisa, Italy*

Abstract

The pattern recognition problem in Ring Imaging CHerenkov (RICH) counters concerns the identification of an unknown number of rings whose centers and radii are assumed to be unknown. In this paper we present an algorithm based on the possibilistic approach to clustering that automatically finds both the number of rings and their position without any *a priori* knowledge. The algorithm has been tested on realistic Monte Carlo LHCb simulated events and it has been shown very powerful in detecting complex images full of rings. The tracking-independent algorithm could be usefully employed after a track based approach to identify remaining trackless rings.

Key words: RICH counters, pattern recognition, possibilistic clustering.
PACS: 29.40.Ka, 29.85.+c, 07.05.Mh

1 Introduction

The LHCb experiment [2] (*A Large Hadron Collider Beauty Experiment for Precision Measurements of CP-Violation and Rare Decays*) at CERN (Geneve, Switzerland) is dedicated to the study of CP-Violation in the B-meson system [1].

* IPHE is now associated to Ecole Polytechnique Federale de Lausanne.
Email address: massone@ge.infm.it (A.M. Massone).

Two Ring Imaging CHerenkov (RICH) counters [3] will carry the very important task of identifying the type of stable charged particles (π, K, p, μ and e) for LHCb.

If a charged particle goes through a dielectric material at a speed greater than the speed of light in this material, photons are emitted at a characteristic angle. This characteristic Cherenkov angle θ_c is given by $\cos\theta_c = 1/(\beta n)$ where βc is the velocity of the charged particle in the medium with index of refraction n. By a clever arrangement of mirrors, the radiated Cherenkov photons are reflected and focused on a detector plane keeping their circular distribution. To optimally carry on the identification task, LHCb is instrumented with two RICHes [7,8]. RICH1 has two radiators (Aerogel with $n = 1.03$ and C_4F_{10} with $n = 1.0014$) while RICH2 has only one radiator (CF_4 with $n = 1.0005$).

In LHCb collaboration an algorithm based on the knowledge of the particle track has been already prepared [4,5] for the Cherenkov rings detection. Its performances are satisfactory except while the information about the tracks are unreliable or completely missed.

In this paper, we present a new approach to the pattern recognition in RICH counters based on the *possibilistic approach to clustering* [6,13]. This algorithm works *without tracking information* and is able to find automatically the number of rings and their position without using information given by the tracking system.

In the next section, relevant aspects of RICH1 and RICH2 for the pattern recognition problem will be summarized. In section 3, we will present the generic clustering problem and we will describe in details the possibilistic algorithm we developed for the pattern recognition problem. In section 4, we will describe the realistic LHCb simulated data used while in section 5 we will develop the application of the clustering algorithm and we will present results. In section 6, a comparison between the possibilistic and the track based algorithm performances will be given. Finally, we will summarize the main results achieved in section 7.

2 RICH1 and RICH2

2.1 Detectors

RICH1 provides the identification of low-momentum tracks [7–9]. The detector is split into two halves on either side of the beam axis. The optical system consists of a single focusing mirror, tilted, reflecting Cherenkov photons onto a

photodetector plane. Unfortunately, for a tilted mirror, the focal surface is no longer spherical and as a result distortions are introduced to the ring image: rings take roughly elliptical shapes. Moreover, the presence of two focusing mirrors, one for each side of the beam pipe, causes that tracks passing close to their interface could generate images in both photodetectors. The detector layout leads to the consequence that the produced rings could be incomplete and then the pattern recognition must be able to recognize not only distorted rings but simple arcs too. From simulation studies [10], for $\beta \approx 1$ tracks (saturated tracks) the mean number of detected photoelectrons is estimated to be 6.9 for the aerogel and 35.3 for the C_4F_{10}. Taking into account that for these tracks Cherenkov angles are about 242 and 53 mrad respectively and the azimuthal distribution of photons is stochastic, a typical event in the RICH1 is a collection of small diameter somewhat densely populated rings from the C_4F_{10} radiator and larger but more sparsely populated rings from the aerogel radiator.

RICH2 provides the identification of high-momentum tracks. The optical system consists of a spherical mirror and a flat mirror for each side of the beam pipe. For a saturated ring, 19.1 detected photoelectrons and a Cherenkov angle of about 32 mrad are expected [10]. As a consequence a typical event in the RICH2 is a collection of only one type rings radiated from the gas.

Obviously, noisy signals are expected such as backscattered charged particles whose signal will be galaxy-like with a radius between about 1/2 (RICH1) and about 1/10 (RICH2) of the typical radius of a regular ring. Another important difference in shape between noisy galaxies and rings is that the first ones are filled with detected photons.

The Monte Carlo studies presented in this paper are related to the state of apparatus as of the Technical Proposal. Since then, an optimization process undertaken by the collaboration and principally aimed to reducing materials in the experiment has slightly changed the typical numbers given above but without significant effects for this presented study.

2.2 Pattern recognition in RICH counters

The pattern recognition problem in RICH counters can be stated as to identify an unknown number of imperfect but roughly elliptical rings made of a low number of discrete hits in presence of background.

An algorithm using the knowledge of the particle track has been developed inside LHCb collaboration [4,5]. It compares the number of detected photoelectrons in the photodetectors with the expected number based on the tracks reconstructed. Assignments to all tracks are treated simultaneously (*global ap-*

proach) and a global log-likelihood fit function minimized. For each track, 5 mass hypothesis can be made (μ, π, K, e, p). For n tracks in a given event the number of possible event hypothesis becomes 5^n. It is therefore infeasible to fully investigate the hypothesis space due to the exponential growth in the number of combinations to be checked. Since the most numerous particles are pions, the global algorithm starts by assuming that all tracks in the event are pions and the log-likelihood for such an event is calculated. For each track the mass hypothesis is then changed and the track is successively supposed to be muon, kaon, electron or proton. The $4n$ new log-likelihood for the $4n$ corresponding hypothesis are recalculated. The mass hypothesis causing the highest increase in the likelihood is accepted and the procedure is repeated until no further improvement in the likelihood is achieved. Doing so, the algorithm converges to the local maxima close to the starting hypothesis. The global algorithm gives good but not perfect results, because it requires to know the reconstructed tracks and hence is sensitive to the tracking system efficiency.

3 Possibilistic Clustering

Clustering is a computer-science and mathematical notion for the broad idea of *grouping similar objects in the same set*. In the frame of Pattern Recognition, such an object is named *data point* and the set is named *class* or *cluster*. The whole set of all data points under consideration is a *database*. The grouping operation aims to recognize common traits shared by the data points of a given class and to group peculiar data points contained in an unlabeled database (training set) into different classes. In the case of pattern recognition in RICH counters, a database is a list of hits in a given photodetector plane, a data point is a single hit and a cluster is a ring. To perform this classification, in this paper, we will present a fully automatic algorithm, a so called *unsupervised* technique. The grouping is obtained via the minimization of a given cost function on the basis of an assigned criteria of *similarity* among data points. The notion of similarity is practically anchored by defining a *distance* between data points.

Supposing that N data points must be classified into C classes, the *Possibilistic Clustering* formulation [6,13] assumes that each data point could belong *simultaneously* to several clusters but with a different *degree of membership*. So it is possible to introduce a *fuzzy membership matrix*, $U = [u_{jk}]$, of dimension $N \times C$, whose elements u_{jk} represent the membership of the k-th data point to the j-th cluster and whose values belong to the continuous range $(0, 1]$. This approach is based on the assumption that the membership value of a data point in a cluster is *absolute* and it does not depend on the membership values of the same data point in any other cluster, in other words *each cluster*

existence is independent of the other ones. The possibilistic constraint is the set of the following conditions:

$$\begin{aligned} & u_{jk} \in [0,1] && \forall j,k \\ & 0 < \textstyle\sum_{k=1}^{N} u_{jk} < N && \forall j \\ & \max_j \ u_{jk} > 0 && \forall k. \end{aligned} \tag{1}$$

The third constraint in Eq. 1 is a relaxation of the probabilistic constraint ($\sum_{j=1}^{C} u_{jk} = 1 \ \forall k$) that would introduce, via the summation, a dependence of membership values on the relative distances among classes — which is a feature that we want to discard in this paper.

By choice, noise figures are not grouped in a dedicated class, hence, a noise data point should have a low membership in all clusters. Representative data points of rings could have high memberships to several clusters non exclusively.

In the next subsection we show the main characteristics of the Possibilistic C-Spherical Shells (PCSS) algorithm [6], introduced in 1993 for the spherical shell clusters detection problem, and we discuss its main aspects. For the RICH rings clustering we had to face some peculiar problems and we developed an algorithm by extending the PCSS. In subsection 3.2 we present the enhancements we added in order to face the particular problem of the Cherenkov rings.

3.1 Possibilistic C-Spherical Shells algorithm

The Possibilistic C-Spherical Shells algorithm searches for cluster prototypes β_j consisting of the couple $(\mathbf{c}_j, r_j)$, where $\mathbf{c}_j$ is the geometrical center and r_j the radius of the j-th ring. The algorithm aims to find a partition of the given training set, by minimizing an *ad hoc* function iteratively.

Given a generic $\mathbf{x}_k$, belonging to a set $\mathbf{Y}$ of N data points, an Euclidean distance measure from the j-th prototype can be defined as:

$$d_{jk}^2 = d^2(\mathbf{x}_k; \mathbf{c}_j, r_j) = (\|\mathbf{x}_k - \mathbf{c}_j\|^2 - r_j^2)^2 \tag{2}$$

The objective function iteratively minimized is:

$$J(U, \mathbf{Y}) = \sum_{j=1}^{C} \sum_{k=1}^{N} u_{jk} d_{jk}^2 + \sum_{j=1}^{C} \eta_j \sum_{k=1}^{N} (u_{jk} \ln u_{jk} - u_{jk}). \tag{3}$$

Table 1
The PCSS Algorithm.

(1) Initialize the number of rings;

(2) Initialize the centers coordinates and radii values;

(3) Estimate η_j parameters values;

(4) **do until** no more prototype changes (within a fixed preset threshold)
   (a) update the centers coordinates and radii values using Eqs. 5 and 6;
   (b) update the membership values of all data points to all rings using Eq. 7;

(5) **end do**

(6) assign the data points with high membership values to the corresponding rings;

(7) do not assign the data points with low membership values to any ring (noise).

where the first term demands the minimization of the distance from the data points to the prototypes (with weighting factors u_{jk}) while the second one, independent of the prototype parameters and the distance measure, is a monotone function in $(0, 1]$ that forces the membership values u_{jk} to be as large as possible in order to avoid the trivial solution $\mathbf{0}$. Finally the η_j parameters must be chosen *a priori* and play a central role; formally they are regularization parameters. They represent the weight of the second term in the objective function with respect to the first one. In order to weight the two terms equally, η_j should be chosen of the order of d_{jk}^2. If clusters with similar distributions are expected, the various η_j should have similar values.

Rewriting the distance (2) in the so called *algebraic form*:

$$d_{jk}^2 = \mathbf{p}_j^{\mathrm{T}} \mathbf{M}_k \mathbf{p}_j + \mathbf{v}_k^{\mathrm{T}} \mathbf{p}_j + b_k \tag{4}$$

where

$$\begin{aligned} & b_k = (\mathbf{x}_k^{\mathrm{T}} \mathbf{x}_k)^2; \qquad \mathbf{v_k} = 2(\mathbf{x}_k^{\mathrm{T}} \mathbf{x}_k)\mathbf{y}_k; \qquad \mathbf{y}_k = \begin{bmatrix} \mathbf{x}_k \\ 1 \end{bmatrix}; \qquad \mathbf{M}_k = \mathbf{y}_k \mathbf{y}_k^{\mathrm{T}} \\ & \mathbf{p}_j = \begin{bmatrix} -2\mathbf{c}_j \\ \mathbf{c}_j^{\mathrm{T}} \mathbf{c}_j - r_j^2 \end{bmatrix} \end{aligned} \tag{5}$$

the vectors $\mathbf{p}_j$ minimizing the objective function (3) are given by:

$$\mathbf{p}_j = -\frac{1}{2}(\mathbf{H}_j)^{-1} \omega_j; \qquad \mathbf{H}_j = \sum_{k=1}^{N} u_{jk} \mathbf{M}_k; \qquad \omega_j = \sum_{k=1}^{N} u_{jk} \mathbf{v}_k \tag{6}$$

while the update equations for u_{jk} are:

$$u_{jk} = \exp\left(-\frac{d_{jk}^2}{\eta_j}\right) \tag{7}$$

Tab. 1 shows the main steps of the basic PCSS algorithm. In particular we want to point out that the initial number of rings has to be (over)estimated. In the possibilistic frame, each prototype is independent of the other ones and then more than one prototype may converge into the same cluster. The identification of identical solutions leads to the automatic determination of the number of clusters present in the database.

3.2 Extended PCSS algorithm

For the RICH rings clustering we developed an algorithm whose core is the PCSS. In particular we embedded the PCSS inside an iterative loop. At each iteration a certain number of rings is detected and data points with a high membership removed from the database. In the following we will describe briefly the additional steps. For a fully detailed description of the whole extended algorithm see [14].

3.2.1 Heuristics for evaluating η_j parameters

For the η_j parameters, we introduced an *ad hoc* definition for the case of spherical shell detection. Interpreting the distance of a generic data point $\mathbf{x}$ from the center of the cluster as function of a radius percentage p ($\|\mathbf{x}-\mathbf{c}_j\|^2 = r_j^2 \pm p r_j^2$) and using Eq. 7, we can link the η_j with the value for which the membership value of a point to a cluster becomes $1/2$. Accordingly, we set:

$$\eta_j = \frac{p^2}{\ln 2} r_j^4. \tag{8}$$

Hence, tuning the p parameter is analogous to controlling the *fuzziness* of the clusters and then the smearing of the hits across the rings. In other words, η_j corresponds to a *zone of influence* of the cluster. Moreover, we inserted the computation of the η_j parameters inside the *do* loop of the PCSS in order to update the corresponding values at each iteration.

3.2.2 Space transformation

We introduced a pre-processing step on the data by a space transformation:

$$\begin{cases} x' = ax + b \\ y' = \frac{a}{\sqrt{1-e^2}} y + \frac{c}{\sqrt{1-e^2}} \end{cases} \tag{9}$$

where e is the expected eccentricity ($e \sim 20\%$) of the detected elliptical rings and a, b and c are scale and translation parameters.

Because of the influence of the parameters η_j, the algorithm is not really sensitive to non-circularity of the figures looked for, but the above transformation reshapes ellipse-like figure in a non-perfect ring which helps the algorithm. Another way would be to change the d_{jk}^2 formula (Eq. 2) to take directly into account the actual elliptic shape. It would have the disadvantage to make Eq. 4 much more complex.

3.2.3 Initialization step

The initialization step is necessary to set an initial overestimated number of rings and their prototypes as starting point. The PCSS is very sensitive to the initialization step, an inaccurate initialization can degrade its performances. We chose the analytic initialization suggested by Muresan in [15].

Given three non collinear points $A(x_1, y_1)$, $B(x_2, y_2)$ and $C(x_3, y_3)$, there is only one ring passing through them. Its center coordinates (x_0, y_0) and radius r are given by the formulas:

$$x_0 = \frac{1}{2} \frac{(x_2^2 - x_3^2 + y_2^2 - y_3^2)(y_1 - y_2) - (x_1^2 - x_2^2 + y_1^2 - y_2^2)(y_2 - y_3)}{(x_2 - x_3)(y_1 - y_2) - (x_1 - x_2)(y_2 - y_3)} \tag{10}$$

$$y_0 = \frac{1}{2} \frac{(x_1^2 - x_2^2 + y_1^2 - y_2^2)(x_2 - x_3) - (x_2^2 - x_3^2 + y_2^2 - y_3^2)(x_1 - x_2)}{(x_2 - x_3)(y_1 - y_2) - (x_1 - x_2)(y_2 - y_3)} \tag{11}$$

$$r = \sqrt{(x_i - x_0)^2 + (y_i - y_0)^2} \qquad i = 1, 2, 3. \tag{12}$$

For each data point in the database, the algorithm looks for the two nearest neighbors and proceeds to (x_0, y_0, r) calculation using Eqs. 10, 11 and 12. Only if r assumes a physically possible value (x_0, y_0, r) is taken as initial prototype.

3.2.4 Cardinality and collapse criteria

After the basic PCSS, with the realistic Monte Carlo data, in the set of reconstructed ring candidates, usually the number of correct ones are less than the expected ones and a certain number of them are completely wrong. To distinguish between good and bad rings, we introduced the so called *Cardinality Criterion*. We have observed that noisy rings are generally less densely populated than the right ones. Via a threshold parameter, an automatic selection of right rings is possible. If the cardinality of a ring is greater than the threshold, the ring is accepted, otherwise it is rejected. Besides that, in the set of accepted rings, the algorithm may find many times the same ring. A *Collapse Criterion* is introduced to identify coincident rings: two rings are identical if their distance in a three-dimensional space $(\mathbf{c}_j, r_j)$ is less than another fixed threshold.

3.2.5 α-cut and stop condition

After the previous double selection, hits belonging to good rings have very high membership values and can be removed from the database if overpassing a threshold, the so called α-cut. Doing so, the database is cleaned from the data points associated to unquestionable rings. In order to find the remaining rings, it is necessary to reiterate the PCSS algorithm on the new *cleaned* database (the number of rings to be found now depends on the surviving hits).

At each step, the algorithm can find a certain number of correct rings and at each step the database cardinality and the number of undetected rings progressively diminish. The algorithm ends up when the remaining patterns are supposed to be outliers (the number of surviving data points is less than some fixed threshold) or when the cardinality threshold, decreased step by step, *reaches a minimal value that we set to 6*. The algorithm is rather sensitive to this value.

Note that the cardinality threshold is progressively decreased to allow the algorithm to find, first of all, well defined rings, and then the less populated ones. In particular, in the last iterations, the cleaned database may consist of rings with holes and arcs that are fragments of rings already recognized and removed. It is worth noting that the algorithm can infer perfectly valid rings from points forming partial rings (arcs).

3.2.6 Filters

At the very end, two filters examine the whole set of recognized rings and eventually reject ghosts or contaminations. One filter rejects rings with points too close in azimuth, the other one rejects rings sharing their whole set of hits

with other valid rings.

4 Application to LHCb Simulated Data

The algorithm depends on 10 critical parameters: number of initial rings, threshold for the cardinality, updating rule for the cardinality threshold, threshold for the collapse, accuracy in the PCSS stop condition, threshold for the α-cut, influence region for the prototypes, minimum number of surviving data points, filter on the azimuthal distribution and filter on shared points.

Hence, the algorithm has to be tuned first by using synthetic databases [11,12], and next by using more realistic data coming from LHCb Monte Carlo simulation. The data refer to the Monte Carlo simulation as of April 2000.

4.1 Data Analysis

We tested the algorithm on the whole database but, in principle, it is an intrinsic limit of the algorithm, to find rings populated by less than 6 hits, in fact, as we explained previously (section 3.2.5) we had to fix a lower limit for the minimal cardinality of the rings to be detected and we set this parameter to that value. Close inspection of Figs. 1(a) and (b), allows to evaluate the percentage of expected rings with more than 6 hits per ring and then the realistic potentialities of the algorithm. In RICH1, from gas 83% of rings verify this condition, while this percentage is only 13% for the aerogel. In RICH2, the distribution of hits per ring is rather uniform (no evident peak), 71% of rings have more than 6 associated hits. Note that the same ring could have several hits in a photodetector plane and very few in the other one. We consider images, from each photodetector plane, separately. As a consequence, a ring, giving less than 6 hits in a photodetector plane (that we can not find), may have a twin ring very populated in the other one (and then detectable).

4.2 Sources of noise

Noise is included in the Monte Carlo (photoelectrons from backward track, incident background photon, photoelectrons/background from incident background non-photon, Rayleigh scattering, backscattered photoelectrons and other stochastic noise). The main source of noise are galaxies-like figures on the detector planes.

Table 2
Efficiency. Values refer to the set of detectable rings, excluding rings with less than 6 hits per ring.

Expected rings in RICH1 (GAS)	312
Detected rings in RICH1 (GAS)	241
Efficiency	77%

Expected rings in RICH2	141
Detected rings in RICH2	141
Efficiency	100%

In a preliminary study [11,12], we have proven that the PCSS algorithm, and the extended ring detector algorithm we prepared, are very robust to the presence of uniform noise (no degradation up to 80% of noise). However their performances could be dramatically degraded by the presence of localized clusters of noise data points. These galaxies, in fact, constitute strong attraction poles that could induce the algorithm to place rings across them. It is then necessary to reject them, at least partially. In a pre-processing step, the algorithm looks for these galaxies, by searching for small areas, characterized by high density of points. This procedure is rather efficient in RICH2 data but not in RICH1 where the overlapping among rings is such that the distinction between high density due to galaxies or ring superposition is ambiguous.

This brief discussion about the strong presence of noise in RICH1 images and previous remarks about the number of hits per ring, leads to an important conclusion: *it is practically impossible, for our algorithm, to detect rings radiated from aerogel*. As we have remarked, only 13% of such rings have more than 6 hits per ring (but at most 10-11) and, furthermore, RICH1 images are generally affected by hundreds of noisy data points, not always rejectable. It is impossible to distinguish a circular long radius shape, made of only 7-8 hits. For these reasons, in the following, we will waive to detect aerogel rings. Undetectable aerogel hits constitute an additional source of noise.

5 Results on RICH Data

In the previous section, we have quantified the percentages of rings effectively detectable by the algorithm. Its performance is then evaluated with respect to these values and then summarized in Tab. 2. Performances are excellent in detecting RICH2 rings, finding all the expected rings (efficiency of 100%). The performances are poorer in RICH1, where the number of detected rings corresponds to an efficiency of 77%.

In RICH2 the algorithm is able to find the correct solution in almost any conditions: huge overlapping, presence of galaxies, few points per ring, presence of arcs instead of complete rings. However, we should not forget that RICH2 data are nearly free of noise and the few present galaxies can be easily rejected.

Table 3
Comparison between Possibilistic and Global algorithm performances. Values refer to the whole set of rings to be found, including rings with less than 6 hits per ring.

	Global	Possibilistic
Lost rings in RICH1 (GAS)	35%	36%
Lost rings in RICH2	29%	29%

Processing RICH1 data, the algorithm faced more difficulties than in RICH2. As previously described, the presence of a high level of noise and its localization in galaxies turned out to be the limiting factor. Analyzing the output of the algorithm, we can say that its performances are still very good in presence of very populated rings, even in presence of overlapping. But it shows problems in recognizing rings with less than 10 hits. Moreover, differently from RICH2 data, the overlapping among rings can be of an unacceptable level for the algorithm.

In Figs. 2, 3 and 4, three examples of clustering are shown. Fig. 2 refers to a RICH2 image while the other ones refer to different difficulty levels in detecting rings in RICH1. All the graphics are plotted in the transformed space (section 3.2.2) in arbitrary units.

6 Possibilistic versus Global algorithm

In the previous section we have shown the algorithm potentialities. We have underlined the quality of the performed clustering in RICH2 counter and highlighted its limits in clustering RICH1 data. But how may it be effectively useful in Pattern Recognition in RICH counters?

The usual algorithm for the Pattern Recognition in RICH counters inside LHCb collaboration is the global approach based on the knowledge of the particle track (section 2.2). This algorithm, however, is not able to detect rings generated by tracks that the tracking system did not reconstruct. It is completely dependent of the tracking system. We postulated that our algorithm could be usefully employed in a second step after the global approach. Hence using track information several rings could be detected, and then removed from the database by the global approach algorithm. At this stage the possibilistic algorithm could detect the remaining ones. For this reason, we have, first, compared the results of the algorithm with the output of the global algorithm, and, then, evaluated its performances when applied in sequence to the global approach.

As we can see in Tab. 3, in RICH1 35% of the rings are associated to tracks

not reconstructed, while for RICH2 the corresponding value is 29%. Using the possibilistic algorithm, we reach values absolutely comparable taking into account the undetectable rings too (the ones with less than 6 hits per ring). It is worth noting that our algorithm reaches these values without using any kind of information about the track and then about the position of centers. Besides that, values for wrong/lost rings from global approach dramatically increase considering the cases in which the particle assignment is wrong. Supposing that the possibilistic algorithm would be applied in a second step, after the global approach, let consider the case in which the global algorithm seems to give the worst results. In Fig. 5(a) the whole database is shown. 45 rings are expected (35 with more than 6 hits) and *PCSS algorithm alone finds 27* when applied to the whole dataset (Fig. 5(b)). *The global algorithm alone detects 17 rings* and misses 28. In Fig. 5(c) we plotted the new database obtained considering only the 28 rings not detected by the global approach and all the noise sources. Among them, 19 have more than 6 points. The possibilistic algorithm applied to this reduced database detects now 16 rings (Fig. 5(d)). Hence, *the two algorithms in sequence detect 33 rings* with more than 6 hits. From this example, it is evident that by using the two algorithms in sequence, it is possible to solve also very complicated cases that, separately, none of them could solve well. Besides that, we would underline, once more, that fragments of very few points in a plane (and then not detectable or lost) could be associated to well populated rings in the other one. Hence, even though we can not detect them in a plane, we can do it in the other one.

7 Conclusions

In this paper, we addressed the pattern recognition problem in RICH counters.

In the case of Cherenkov rings detection we have presented results using a clustering algorithm, whose core is the Possibilistic C-Spherical Shell algorithm, to recognize rings on images without any preliminary knowledge of number and position of the rings. The algorithm has been shown very powerful in detecting complex images full of rings. Besides that, the rings are not requested to be complete, only arcs are sufficient to recognize the underlying rings. The fuzziness included in the algorithm adds tolerance to the imperfect circular shape of the rings and to the intrinsic scattering of Cherenkov photons. The algorithm has been tested on realistic Monte Carlo LHCb simulated events. Its performances are excellent in detecting RICH2 rings. It is able to find the correct solution in almost any conditions: huge overlapping, presence of galaxies, few hits per ring, presence of arcs instead of complete rings. Processing RICH1 data, the first conclusion we have made is that because of the strong presence of noise in RICH1 images and low number of hits per ring, it is impossible, for the proposed algorithm, to detect rings from aerogel radi-

ator. On the contrary, its performances are still very good in detecting rings from gas radiator, even in presence of overlap. Anyway the presence of a high level of noise and its localization in galaxies, turned to be a limiting factor in RICH1 images. The usual algorithm for the Pattern Recognition in RICH counters inside LHCb collaboration is the so called global approach based on the knowledge of the particle track. This algorithm, however, is not able to detect rings generated by tracks that the tracking system did not reconstruct. We have shown that the algorithm described in this paper could be usefully employed in a second step after the global approach. Hence, using track information several rings could be detected by the global algorithm, and then removed from the database. At this stage the extended PCSS algorithm could detect the remaining ones.

References

[1] J.H. Christenson et al., *Phys. Rev. Lett.* 13, 138 (1964).

[2] The LHCb Coll., *LHCb Technical Proposal*, CERN/LHCC/98-4.

[3] T. Ypsilantis and J. Seguinot, *Nucl. Instr. Meth. A* 343, 30 (1994).

[4] R. Forty, *Ring-Imaging Cherenkov Detectors for LHC-B.* LHC-B/96-5 RICH/96-1.

[5] R. Forty and O. Schneider, *RICH pattern recognition.* LHCB/98-040 RICH.

[6] R.Krishnapuram and J.M. Keller. *A Possibilistic Approach to Clustering.* IEEE Transactions on Fuzzy Systems, 1(2) pp.98–110. (1993).

[7] G.J. Barber et al., *The mechanical design of RICH-1.* LHCb 2000-077 RICH

[8] LHCb RICH Group, *LHCb RICH 2 Engineering Design Review Report.* LHCb EDR 2002-009.

[9] The LHCb Coll., *LHCb RICH Technical Design Report.* CERN/LHCC 2000-037, LHCb TDR 3, 7 September 2000.

[10] M. Adinolfi et al., *A simulation study of the LHCb RICH performance.* LHCb 2000-066 RICH.

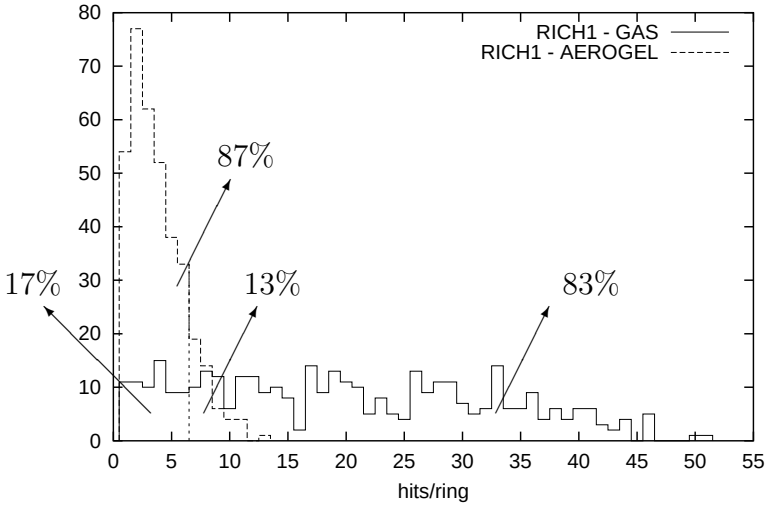
[11] A.M.Massone, L.Studer and F.Masulli. *Pattern Recognition in RICH Counters using the Possibilistic C-Spherical Shell Algorithm.* Proceedings of the Fourth International Conference on Knowledge-Based Intelligent Engineering Systems & Allied Technologies (KES'2000) Brighton, UK. pp. 792-795. IEEE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

[12] A.M.Massone, L.Studer and F.Masulli. *Possibilistic Rings Detection for RICH Pattern Recognition.* Proceedings of the Eusflat 2001 International Conference in Fuzzy Logic and Technology. pp.63-66. 2001

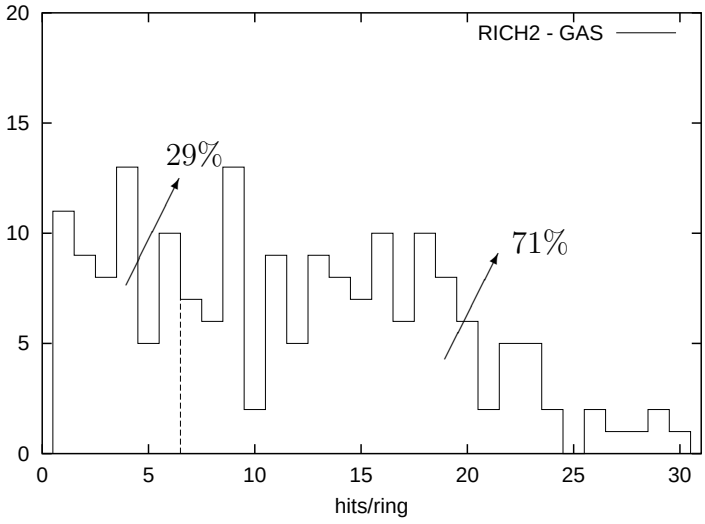
[13] R. Krishnapuram and J.M. Keller. *The Possibilistic C-means Algorithm: Insights and Recommendations.* IEEE Transactions on Fuzzy Systems, 4(3) pp.385–393. (1996).

[14] A.M.Massone. *A study of soft computing clustering methods with illustrative applications in segmenting MR images and detecting trackless ring for RICH detectors.* PhD Thesis. Université de Lausanne. Lausanne. Switzerland. http://lphe.epfl.ch/engl/publications/theses.html#these

[15] L. Muresan et al. *Deformable Templates for Circle Recognition.* Proceedings of *Computer Modelling and Computing in Physics conf.*, Dubna Sept. 16-21 1996, Editors: A.Fedorov, R.Shakhbaghyan, E.Zhidkov, JINR publ. D5-11-97-112, Dubna, 1997, 214-218.



(a)



(b)

Fig. 1. (a) RICH1. Histogram of the number of hits per ring from GAS and AEROGEL. Total rings are 377 for GAS and 363 for AEROGEL. (b) RICH2. Histogram of the number of hits per ring from GAS. Total rings are 200.

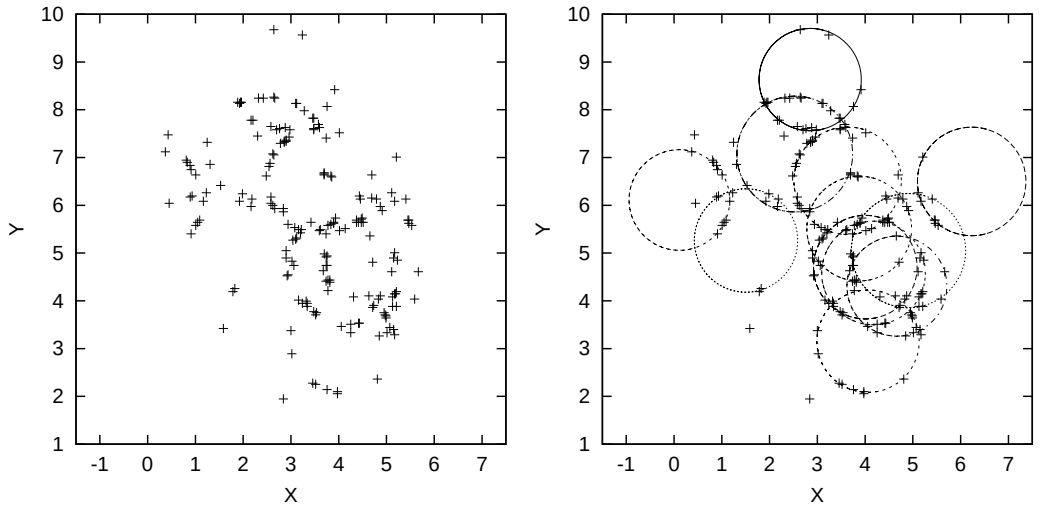


Fig. 2. RICH2. (a) Input database. (b) Solution found superimposed to the database. In this example 13 rings were expected, but only 12 with more than 6 points. The algorithm finds all the 12 well populated rings. The quality of the solution is evident, especially in the region of high overlapping and in the border regions where hits describe arcs and not complete rings.

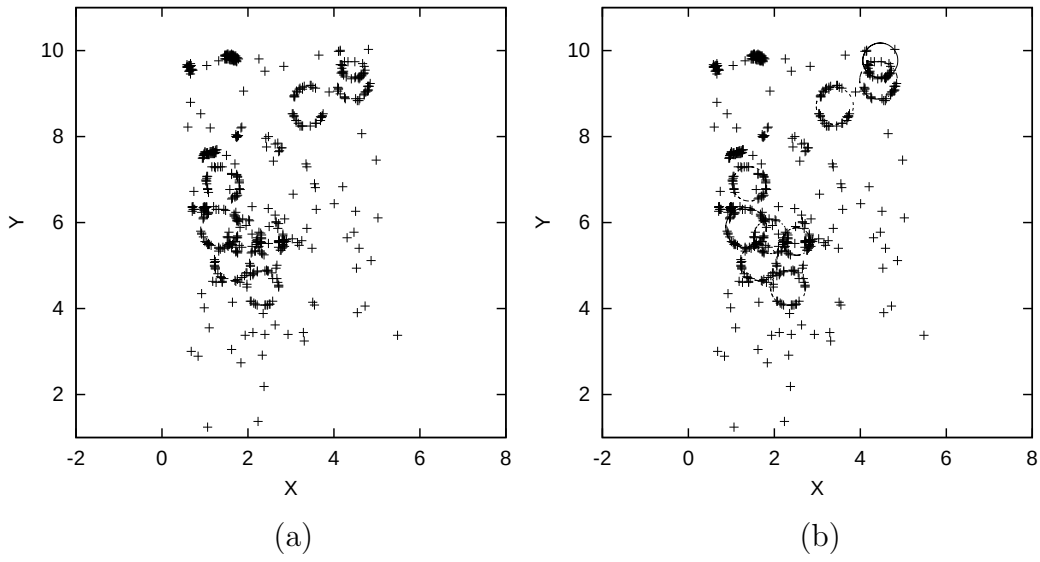
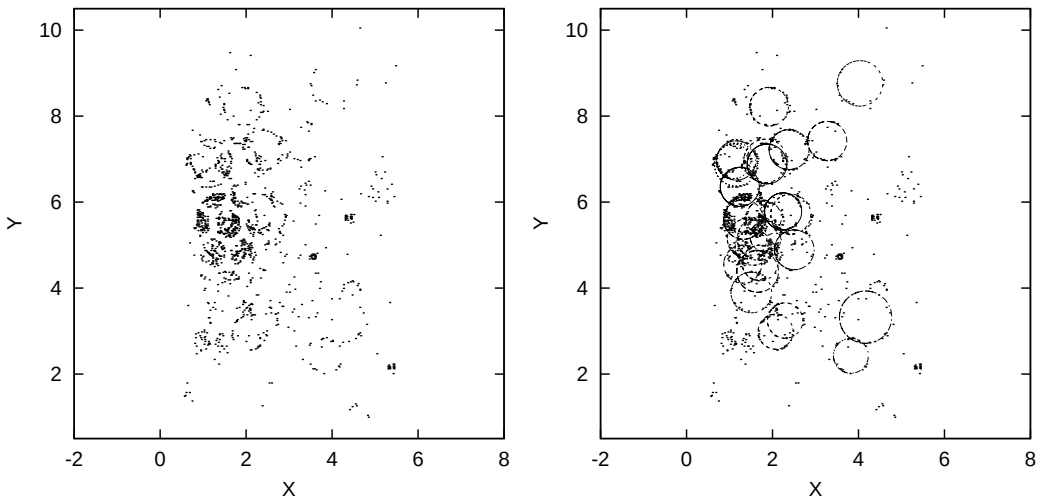


Fig. 3. RICH1. (a) Input database. (b) Solution found. The expected rings are 10, while the algorithm finds 9. In this simple example, there is no high overlapping among rings, but we can appreciate how the algorithm is able to approximate the ellipses by rings, in the upper part of the image (b). We can also appreciate the presence of two big *galaxies* in the left upper part of the figures and a general level of noise higher than the one of Fig. 2.



(a) (b)

Fig. 4. RICH1. (a) Input database. (b) Solution found. This example is a very complicated case with high overlap and high number of rings to be found. The overlap is huge and it seems impossible, in some regions, to distinguish different circular shapes. The expected rings were 42 but the algorithm finds 33. The lost rings are populated, in most cases, by less than 10 hits.

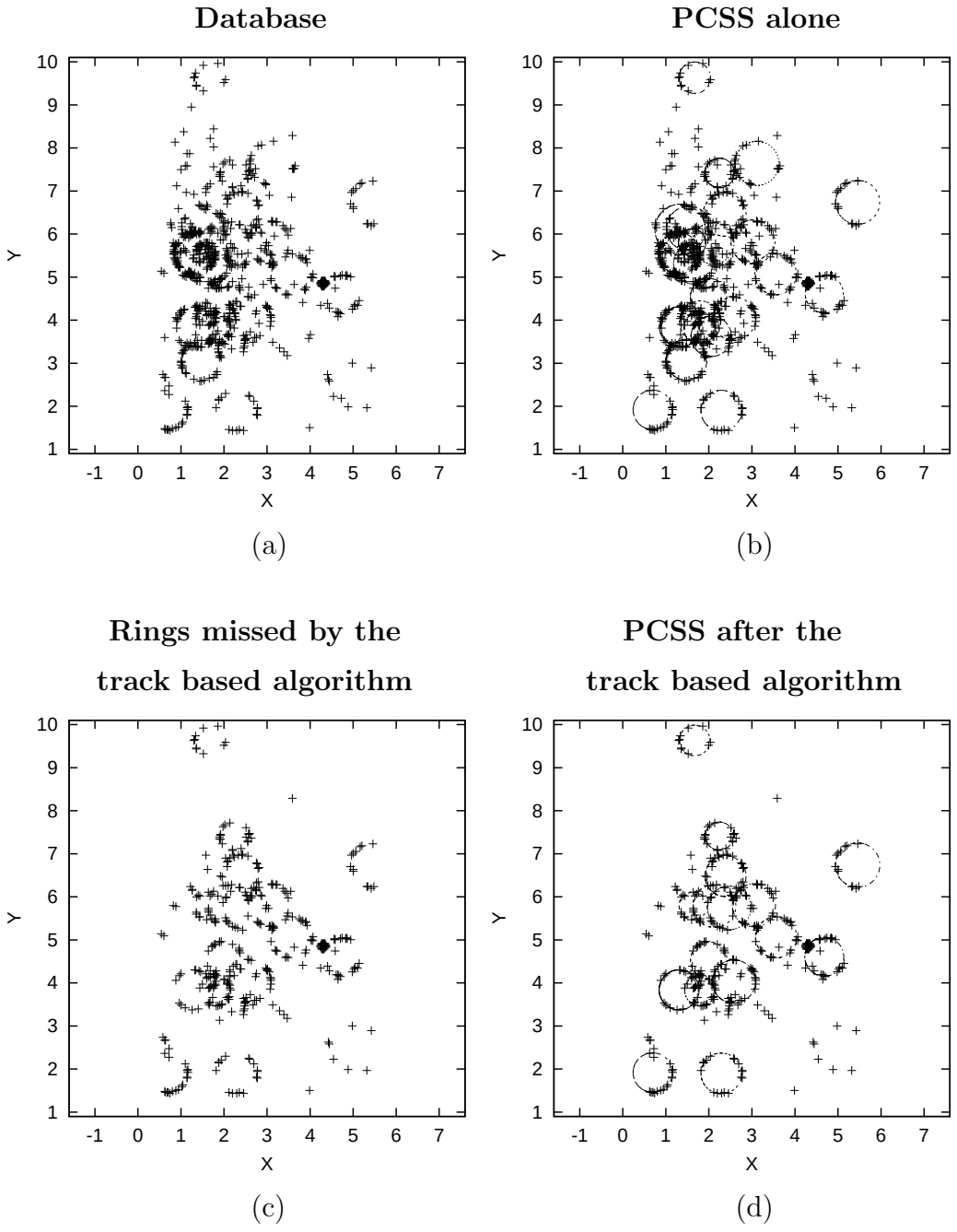


Fig. 5. The PCSS algorithm may be usefully applied in sequence after the track based algorithm.