Unsupervised Categorization for Image Database Overview

Bertrand Le Saux and Nozha Boujemaa

INRIA, Imedia Research Group, BP 105, F-78153 Le Chesnay, France,
`Bertrand.Le-Saux@inria.fr`,
WWW home page: `http://www-rocq.inria.fr/ lesaux`

Abstract. We introduce a new robust approach to categorize image databases : Adaptative Robust Competition (ARC). Providing the best overview of an image database helps users browsing large image collections. Estimating the distribution of image categories and finding their most descriptive prototype represent the two main issues of image database categorization. Each image is represented by a high-dimensional signature in the feature space. A principal component analysis is performed for every feature to reduce dimensionality. Image database overview by categorization is computed in challenging conditions since clusters are overlapping and the number of clusters is unknown. Clustering is performed by minimizing a Competitive Agglomeration objective function with an extra noise cluster collecting outliers.

1 Introduction

Over the last few years, partly due to the development of the Internet, more and more multimedia documents that include digital images have been produced and exchanged. However, locating a target image in a large collection became a crucial problem. The usual way to solve it consists in describing images by keywords. Because this is a human operation this method suffers from subjectivity and text ambiguity and requires huge time to manually annotate a whole database. By image analysis images can be indexed by automatic description which only depend on their objective visual content. So Content-based Image Retrieval (CBIR) became a highly active research field.

The usual scenario of CBIR is a query by example, which consists in retrieving images of the database similar to a given one. The purpose of browsing is to help the user finding his image query by providing first the best overview of the database. Since the database cannot be presented entirely, a limited number of key images have to be chosen. It means we have to find the most informative images which allow the user to know what the database contains. The main issue is to estimate the distribution (usually multi-modal) of image categories. Then we need the most representative image for each category.

Practically, this is a critical point in the scenario of content-based query by example: the “page zero” problem. Existing systems often begin by presenting either randomly chosen images or keywords. In the first case, some categories are missed, and some images can be visually redundant. The user has to pick several random subsets to find an image corresponding to the one he has in mind. Only then can the query by example be performed. In the second case, images are manually annotated with keywords, and the first query is processed using keywords. Thus there is a need for presenting a summary of the database to the user.

A popular way to find partitions in complex data is prototype-based clustering algorithm. The fuzzy version (Fuzzy C-Means [1]) has been constantly improved for twenty years by the use of the Mahalanobis distance [2], the adjunction of a noise cluster [3] or the competitive agglomeration algorithm [4] [5]. A few attempts to organize and browse image databases have been made: Brunelli and Mich [6], Medasani and Krishnapuram [7] and Frigui et al. [8]. A key point of categorization is the input data representation. A set of signatures (color, texture and shape) allows to describe the visual appearance of the image. The content-based categorization should be performed by clustering these signatures. This operation is computed in challenging conditions. The feature space is high-dimensional: computations are affected by the curse of dimensionality. The number of clusters in the image database is unknown. Natural categories have various shapes (sometimes hyper-ellipsoidal but often more complex), they are overlapping and they have various densities.

The paper is organized as follows: §2 presents the background of our work. Our method is then presented in section 3. The results on image databases are discussed and compared with other clustering methods in section 4 and section 5 summarizes our concluding remarks.

2 Background

The Competitive Agglomeration (CA) algorithm [4] is a fuzzy partitional algorithm which allows not to specify the number of clusters. Let $X = \{x_i| \; i \; \epsilon \; \{1,..,N\}\}$ be a set of N vectors representing the images. Let $B = \{\beta_j| \; j \; \epsilon \; \{1,..,C\}\}$ represents prototypes of the C clusters. Competitive Agglomeration (CA) algorithm minimizes the following objective function:

$$J = \sum_{j=1}^{C} \sum_{i=1}^{N} (u_{ji})^2 d^2(x_i, \beta_j) - \alpha \sum_{j=1}^{C} \Big[\sum_{i=1}^{N} (u_{ji}) \Big]^2 \tag{1}$$

Constrained by:

$$\sum_{j=1}^{C} u_{ji} = 1, for \; i \; \epsilon \{1,..,N\} \tag{2}$$

$d^2(x_i, \beta_j)$ represents the distance from an image signature x_i to a cluster prototype β_j. The choice of the distance depends on the type of clusters having to be detected. For spherical clusters, Euclidean distance will be used. u_{ji} is the membership of x_i to a cluster j.

The first term is the standard FCM objective function [1]: the sum of weighted square distances. It allows us to control shape and compactness of clusters. The second term (the sum of squares of clusters' cardinalities) allows us to control the number of clusters. By minimizing both these terms together, the data set will be partitioned in the optimal number of clusters while clusters will be selected to minimize the sum of intra-cluster distances.

The cardinality of a cluster is defined as the sum of the memberships of each image to this cluster:

$$N_s = \sum_{i=1}^{N} (u_{si}) \tag{3}$$

Membership can be written as:

$$u_{st} = u_{st}^{FCM} + u_{st}^{Bias}, \tag{4}$$

where:

$$u_{st}^{FCM} = \frac{[1/d^2(x_t, \beta_s)]}{\sum_{j=1}^{C} [1/d^2(x_t, \beta_j)]}, \tag{5}$$

and:

$$u_{st}^{Bias} = \frac{\alpha}{d^2(x_t, \beta_s)} (N_s - \frac{\sum_{j=1}^{C} [1/d^2(x_t, \beta_j)] N_j}{\sum_{j=1}^{C} [1/d^2(x_t, \beta_j)]}) \tag{6}$$

The first term in equation (4) is the membership term in FCM algorithm and takes into account only relative distances to the clusters. The second term is a bias term which is negative for low cardinality cluster and positive for strong clusters. This bias term leads to a reduction of cardinality of spurious clusters which are discarded if their cardinality drops below a threshold. As a result only good clusters are conserved.

α should provide a balance [4] between the two terms of (1) so α at iteration k is defined by :

$$\alpha(k) = \eta_0 \exp(-k/\tau) \frac{\sum_{j=1}^{C} \sum_{i=1}^{N} (u_{ji})^2 d^2(x_i, \beta_j)}{\sum_{j=1}^{C} \left[\sum_{i=1}^{N} (u_{ji}) \right]^2} \tag{7}$$

α is weighted by a factor which decreases exponentially along iterations. In the first iterations the second term of equation (1) dominates so the number of clusters drops rapidly. Then, when the optimal number of clusters is found, the first term dominates and the CA algorithm seeks the best partition of the signatures.

3 Adaptative Robust Competition (ARC)

3.1 Dimensionality reduction

A signature space has been built for a 1440 image database (Columbia Object Image Library [9]). It contains 1440 gray scale images representing 20 objects, where each object is shot every 5 degrees. This feature space is high-dimensional and contains three signatures:

1. Intensity distribution (16-D): the gray level histogram.

2. Texture (8-D): the Fourier power spectrum is used to describe the spatial frequency of the image [10].
3. Shape and Structure (128-D): the correlogram of edge-orientations histogram (in the same way as color correlogram presented at [11]).

The whole space is not necessary to distinguish images. To prevent clustering from expensive computation, a principal component analysis is performed to reduce the dimensionality. For each feature only the first main components are kept.

To visualize the problems raised by the categorization of image databases, the distribution of image signatures is shown on figure 1. This figure presents the subspace corresponding to the three principal components of the feature gray level histogram. Each natural category is represented with a different color. Two main problems appear: categories overlap and natural categories have different and various shapes.

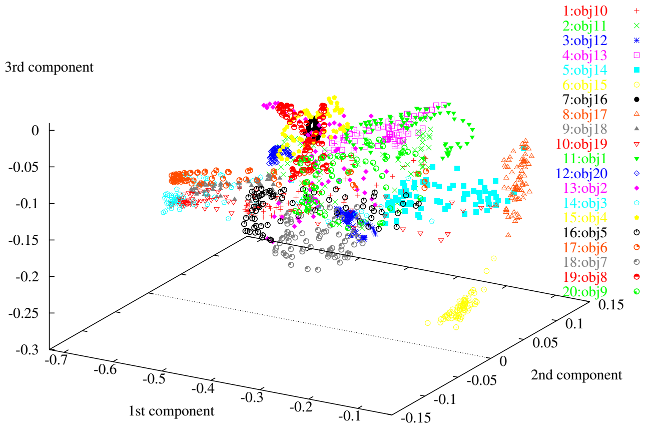

Fig. 1. Distribution of gray level histograms for Columbia database on the three principal components

3.2 Adaptative competition

α is the weighting factor of the competition process. In equation (7) α is chosen according to the objective function and has the same value and effect for each cluster. Though, during the process, α influences the computation of memberships in equations (4) and (6). The term u_{st}^{Bias} appreciates or depreciates the membership u_{st} of data point x_t to cluster t according to the cardinality of the cluster. This will cause this cluster to be conserved or discarded respectively.

Since clusters may have different compactness, the problem is to attenuate the effect of u_{st}^{Bias} for loose clusters, in order to not discard them too rapidly. We introduce an average distance for each cluster s:

$$d^2_{moy}(s) = \frac{\sum_{i=1}^{N}(u_{si})^2 d^2(x_i, \beta_s)}{\sum_{i=1}^{N}(u_{si})^2} \;\; for\, 1 \leq s \leq C \tag{8}$$

And an average distance for the whole set of image signatures :

$$d^2_{moy} = \frac{\sum_{j=1}^{C}\sum_{i=1}^{N}(u_{ji})^2 d^2(x_i, \beta_j)}{\sum_{j=1}^{C}\sum_{i=1}^{N}(u_{ji})^2} \tag{9}$$

Then, α in equation (6) is expressed as:

$$\alpha_s(k) = \frac{d^2_{moy}}{d^2_{moy}(s)}\alpha(k) \;\; for\, 1 \leq s \leq C \tag{10}$$

The ratio $d^2_{moy}/d^2_{moy}(s)$ is lower to 1 for loose clusters, so the effect of u_{st}^{Bias} is attenuated : cardinality of cluster is slowly reduced. On the contrary, $d^2_{moy}/d^2_{moy}(s)$ is greater than 1 for compact clusters, so both memberships to these clusters and cardinalities are increased: they are more resistant in the competition process. Hence we build an adaptative competition process given by $\alpha_s(k)$ for each cluster s.

3.3 Robust clustering

A solution to deal with noisy data and outliers is to capture all the noise signatures in a single cluster [3]. A virtual noise prototype is defined, which is always at the same distance δ from every point in the data-set. Let this noise cluster be the first cluster, and noise prototype noted as β_1. So we have:

$$d^2(x_i, \beta_1) = \delta^2 \tag{11}$$

Then the objective function (1) has to be minimized with the following particular conditions:

- Distances for the good clusters j are defined by:

  $$d^2(x_i, \beta_j) = (x_i - \beta_j)^T A_j (x_i - \beta_j) \;\; for\, 2 \leq \; j \; \leq C. \tag{12}$$

  where A_j are positive definite matrices. If A_j are identity matrix, then the distance is Euclidean distance, and the prototypes of clusters j for $2 \leq \; j \; \leq C$ are:

  $$\beta_j = \frac{\sum_{i=1}^{N}(u_{ji})^2 x_i}{\sum_{i=1}^{N}(u_{ji})^2} \tag{13}$$

- For the noise cluster $j = 1$, distance is given by (11).

The noise distance δ has to be specified. It would vary from an image database to another, so it would be based on data-set statistical information. It is computed as the average distance between image signatures and good cluster prototypes:

$$\delta^2 = \delta_0^2 \frac{\sum_{j=2}^{C} \sum_{i=1}^{N} d^2(x_i, \beta_j)}{N(C-1)} \tag{14}$$

The noise cluster is then supposed to catch outliers that are at an equal mean distance from all cluster prototypes. Initially, δ cannot be computed using this formula, since distances are not yet computed. It is just initialized to δ_0, and the noise cluster becomes significant after a few iterations. δ_0 is a factor which can be used to enlarge or minimize the size of the noise cluster, though in the results that will be presented, $\delta_0 = 1$.

The new ARC algorithm using adaptative competitive agglomeration and noise cluster can now be summarized:

```
Fix the maximum number of clusters $C$.
Initialize randomly prototypes for $2 \leq j \leq C$.
Initialize memberships with equal probability for each image to belong to each cluster.
Compute initial cardinalities for $2 \leq j \leq C$ using equation (3).
**Repeat**
    Compute $d^2(x_i, \beta_j)$ using (11) for $j = 1$ and (12) for $2 \leq j \leq C$.
    Compute $\alpha_j$ for $1 \leq j \leq C$ using equations (10) and (7).
    Compute memberships $u_{ji}$ using equation (4) for each cluster and each signature.
    Compute cardinalities $N_j$ for $2 \leq j \leq C$ using equation (3).
    For $2 \leq j \leq C$, if $N_j < threshold$, discard cluster $j$.
    Update number of clusters $C$.
    Update prototypes using equation (13).
    Update noise distance $\delta$ using equation (14).
**Until** (prototypes stabilized).
```

Hence a new clustering algorithm is proposed. The two next points address two problems raised by image database categorization.

3.4 Choice of distance for good clusters

What would be the most appropriate choice for (12) ? The image signatures are composed of different features which describe different attributes. The distance between signatures is defined as the weighted sum of partial distances for each feature $1 \leq f \leq F$:

$$d(x_i, \beta_j) = \sum_{f=1}^{F} w_{j,f} d_f(x_i, \beta_j) \tag{15}$$

For each feature, the natural categories in image databases have various shapes, the more often hyper-ellipsoidal, and overlap each other. To retrieve such clusters, Euclidean distance is not appropriate. So the Mahalanobis distance [2] is used to discriminate image signatures. For clusters $2 \leq j \leq C$, partial distances for feature f are computed using :

$$d_f(x_i, \beta_j) = |C_{j,f}|^{1/p_f}(x_{i,f} - \beta_{j,f})^T C_{j,f}^{-1}(x_{i,f} - \beta_{j,f}) \tag{16}$$

where $x_{i,f}$ and $\beta_{j,f}$ are the restrictions of image signature x_i and cluster prototype β_j to the feature f. p_f is the dimension of both $x_{i,f}$ and $\beta_{j,f}$: it is the dimension of the subspace corresponding to feature f. $C_{j,f}$ is the covariance matrix (of dimension $p_f \times p_f$) of cluster j for the feature f:

$$C_{j,f} = \frac{\sum_{i=1}^{N}(u_{ji})^2(x_{i,f} - \beta_{j,f})(x_{i,f} - \beta_{j,f})^T}{\sum_{i=1}^{N}(u_{ji})^2} \tag{17}$$

3.5 Normalization of features

The problem is to compute the weights $w_{j,f}$ used in equation (15). The features have different orders of magnitude and different dimensions, so the distance over all features cannot be defined as a simple sum of partial distances for each feature. The idea is to learn the weights during the clustering process. Ordered Weight Averaging [12] is used, as proposed in [8].

First, partial distances are sorted in ascending order. For each feature f, the rank of corresponding partial distance is obtained:

$$r_f = rank(d_f(x_i, \beta_j)) \tag{18}$$

And the weight at iteration $k > 0$ is updated using:

$$w_{j,f}^{(k)} = w_{j,f}^{(k-1)} + \frac{2(F - r_f)}{F(F+1)} \tag{19}$$

It has two positive effects. First, features with small values are weighted with a higher weight than those with large values, so the sum of partial distances is equilibrated. Secondly, since the weights are computed during the clustering process, if some images are found to be similar according to one feature, their partial distance will be small, and the effect of this feature will be accentuated: it allows to find a cluster which contains images similar according to a single main feature.

3.6 Algorithm outline

Fix the maximum number of clusters C.
Initialize randomly prototypes for $2 \leq j \leq C$.
Initialize memberships with equal probability for each image to belong to each cluster.
Initialize feature weights uniformly for each cluster $2 \leq j \leq C$.

```
Compute initial cardinalities for $2 \leq j \leq C$.
Repeat
    Compute covariance matrix for $2 \leq j \leq C$ and feature subsets $1 \leq f \leq F$ using (17).
    Compute $d^2(x_i, \beta_j)$ using (11) for $j = 1$ and (16) for $2 \leq j \leq C$.
    Update weights for clusters $2 \leq j \leq C$ using (19) for each feature.
    Compute $\alpha_j$ for $1 \leq j \leq C$ using equations (10) and (7).
    Compute memberships $u_{ji}$ using equation (4) for each cluster and each signature.
    Compute cardinalities $N_j$ for $2 \leq j \leq C$.
    For $2 \leq j \leq C$, if $N_j < threshold$ discard cluster $j$.
    Update number of clusters $C$.
    Update prototypes using equation (13).
    Update noise distance $\delta$ using equation (14).
Until (prototypes stabilize).
```

4 Results and discussion

The ARC algorithm is compared with two other clustering algorithms: the basic CA algorithm [4] and the Self-Organization of Oscillator Network (SOON) algorithm [8].

The SOON algorithm can be summarized as follows:

1. Each image signature is associated to an oscillator characterized by a phase variable that belongs to $[0, 1]$.
2. Whenever an oscillator's phase reaches 1, it resets to 0 and other oscillators' phases are either increased or decreased according to a similarity function.
3. Oscillators begin to clump together in small groups. Within each group, oscillators are phase-locked. After a few cycles, existing groups get bigger by absorbing other oscillators and merging with other groups.
4. Eventually, the system reaches a stable state where the image signatures are organized into the optimal number of stable groups.

For each category, a prototype is chosen according to the following steps:

- The average value of each feature is computed over image.
- Then, the average of all images defines a virtual prototype.
- The real prototype is the nearest image to the virtual one.

The ground truth of Columbia database is shown on figure 2. The three summaries are presented on figures 2 and 3. Quite all the natural categories are retrieved with the three methods. But with SOON or CA algorithms, some categories are split in several clusters, so several prototypes are redundant. Our method provides a better summary with less redundancy.

Tables 1 and 2 present the membership matrices of objects to clusters which describe the content of each cluster. Since the simple CA algorithm has no cluster to collect ambiguous image signatures, clusters obtained with this method are noisy. Besides

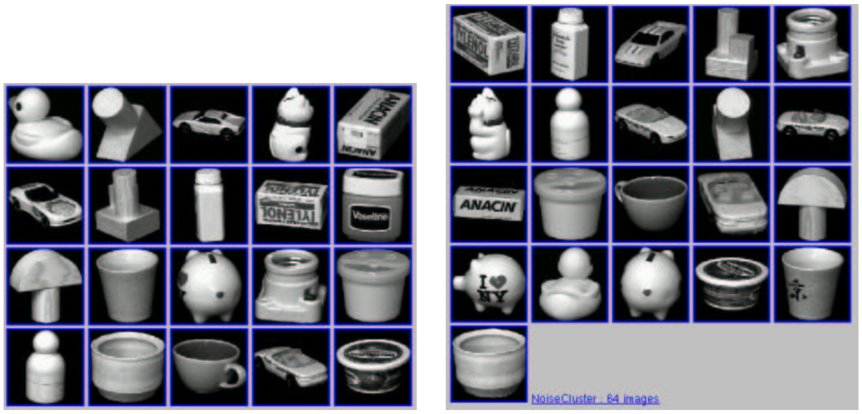

Fig. 2. left: ground truth: the 20 objects of the Columbia database, right: Summary obtained with ARC algorithm

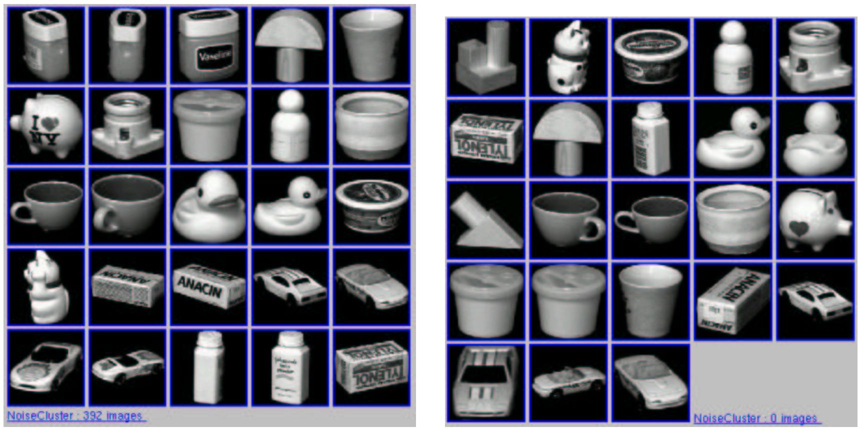

Fig. 3. left: Prototypes of clusters obtained with SOON algorithm, right: Prototypes of clusters obtained with CA algorithm

Table 1. This matrix shows how many pictures of each object belong to a cluster obtained with ARC.

Object	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Cluster																				
1	72	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
2	.	3	1	1	.	.	.	.	.	.	2	.	3	.	.	.	.	.	.	.
3	.	.	48	.	4	4	.	.	.	5	.	.	.	.	.	.	.	.	4	.
4	.	3	4	70	.	.	.	15	.	.	.	.	.	.	.	13	.	.	.	.
5	.	.	.	.	32	.	.	.	.	.	1	.	.	.	.	.	.	.	.	.
6	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
7	.	.	.	.	3	.	67	.	.	.	12	.	.	.	.	.	.	.	.	.
8	.	.	.	.	2	.	5	57	.	.	1	.	.	.	.	.	.	.	.	.
9	.	.	.	.	13	.	.	.	70	5	.	.	.	.	.	.	.	.	.	.
10	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
11	.	9	.	.	.	.	.	.	.	1	51	.	.	.	.	.	.	.	.	.
12	.	.	.	.	3	.	.	.	.	5	.	72	.	.	.	.	.	.	.	.
13	.	22	.	.	.	.	.	.	.	.	5	.	21	.	.	.	.	.	.	.
13	.	12	.	.	.	.	.	.	.	.	.	.	48	.	.	.	.	.	.	.
14	.	.	.	.	.	1	.	.	.	.	.	.	.	72	.	.	.	.	1	.
15	.	.	.	.	.	.	.	.	.	.	.	.	.	.	72	.	.	.	.	.
16	.	.	.	.	.	2	.	.	.	.	.	.	.	.	.	59	.	.	.	.
17	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	72	.	.	.
18	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	72	.	.
19	.	.	18	.	2	35	.	.	.	14	.	.	.	.	.	.	.	.	26	.
19	.	.	.	1	2	16	.	.	.	16	.	.	.	.	.	.	.	.	23	.
19	.	.	11	.	1	14	.	.	.	.	.	.	.	.	.	.	.	.	19	.
20	.	.	.	.	.	.	.	.	.	2	.	.	.	.	.	.	.	.	.	72
noise	.	23	5	.	10	.	.	.	2	24	.	.	.	.	.	.	.	.	.	.

Table 2. The left matrix shows how many pictures of each object belong to a cluster obtained with CA and the right matrix shows the result of the same experiment with SOON.

Object	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Cluster																				
1	42	.	.	4	.	.	.	1	.	2	6	.	.	.	.	.	.	.	.	.
1	30	.	.	.	.	.	.	9	.	.	1	.	.	.	.	.	.	.	.	.
2	.	35	.	.	.	.	3	1	.	.	1	.	.	.	.	.	.	.	.	.
3	.	.	8	.	.	30	.	.	.	.	.	.	.	.	.	.	.	.	26	.
3	.	.	10	.	.	.	.	.	.	1	.	.	.	.	.	.	.	.	10	.
4	.	1	2	31	22	.	.	1	3	3	.	.	.	.	.	.	.	.	.	.
5	.	.	.	.	10	.	5	.	.	54	3	.	.	.	.	.	.	.	.	.
6	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
7	.	.	.	.	1	.	61	.	.	.	.	.	.	.	.	.	.	14	.	.
8	.	.	.	.	2	.	.	21	19	.	.	.	.	.	.	.	.	.	.	44
9	.	.	.	.	5	.	.	19	47	.	.	.	.	.	.	.	.	.	.	.
10	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
11	.	5	.	.	1	.	.	3	.	.	49	.	.	.	.	.	.	.	.	.
12	.	.	.	.	12	.	.	.	.	.	.	72	.	.	.	.	.	.	.	.
13	.	17	.	.	.	.	.	.	.	.	6	.	72	.	.	.	.	.	.	.
14	.	.	.	.	.	.	.	6	.	.	.	.	.	72	.	.	.	.	.	.
15	.	.	.	.	.	.	1	.	.	.	.	.	.	.	33	.	.	.	.	.
15	.	.	.	.	.	.	2	.	.	.	4	.	.	.	39	.	.	.	.	.
16	.	13	.	37	.	.	.	12	.	.	2	.	.	.	.	72	.	.	.	.
17	.	.	.	.	1	.	.	.	.	.	.	.	.	.	.	.	72	.	.	.
18	.	.	.	.	10	.	.	.	.	3	.	.	.	.	.	.	.	29	.	.
18	.	.	.	.	.	.	.	.	.	1	.	.	.	.	.	.	.	29	.	.
19	.	.	40	.	8	25	.	.	.	8	.	.	.	.	.	.	.	.	26	.
19	.	.	12	.	.	17	.	.	.	.	.	.	.	.	.	.	.	.	10	.
20	.	.	.	.	.	.	.	.	3	.	.	.	.	.	.	.	.	.	.	28

Object	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Cluster																				
1	21	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
1	51	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
2	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
3	.	.	7	.	.	6	.	.	.	.	.	.	.	.	.	.	.	.	.	.
4	.	.	.	72	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
5	.	.	.	.	15	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
5	.	.	.	.	19	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
6	.	.	4	.	.	5	.	.	.	.	.	.	.	.	.	.	.	.	6	.
6	.	.	40	.	.	43	.	.	.	.	.	.	.	.	.	.	.	.	42	.
7	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
8	.	.	.	.	.	.	.	16	.	.	.	.	.	.	.	.	.	.	.	.
8	.	.	.	.	.	.	.	40	.	.	.	.	.	.	.	.	.	.	.	.
9	.	.	.	.	.	.	.	.	14	.	.	.	.	.	.	.	.	.	.	.
10	.	.	.	.	.	.	.	.	.	10	.	.	.	.	.	.	.	.	.	.
10	.	.	.	.	.	.	.	.	.	16	.	.	.	.	.	.	.	.	.	.
10	.	.	.	.	.	.	.	.	.	10	.	.	.	.	.	.	.	.	.	.
11	.	.	.	.	.	.	.	.	.	.	26	.	.	.	.	.	.	.	.	.
12	.	.	.	.	.	.	.	.	.	.	.	72	.	.	.	.	.	.	.	.
13	.	.	.	.	.	.	.	.	.	.	.	.	13	.	.	.	.	.	.	.
14	.	.	.	.	.	.	.	.	.	.	.	.	.	71	.	.	.	.	.	.
15	.	.	.	.	.	.	.	.	.	.	.	.	.	.	72	.	.	.	.	.
16	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	72	.	.	.	.
17	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	72	.	.	.
18	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	39	.	.
18	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	33	.	.
19	.	.	2	.	.	3	.	.	.	.	.	.	.	.	.	.	.	.	5	.
20	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	72
noise	.	72	19	.	38	15	72	16	57	36	46	.	.	1	.	.	.	.	19	.

the main natural category retrieved in a cluster, there are always other images which belong to a neighbor cluster or to a wide spread cluster.

This problem is solved with both other methods. With ARC or SOON algorithms, more than a third of categories are perfectly clustered, i.e. all the images of a single category are grouped in a single cluster. The other natural categories present more variation among their images, so are more difficult to retrieve.

Let's consider one of these categories : the images representing the drug package 'tylenol'. It presents several difficulties: it is wide spread, and another category which represents another drugs package is very similar. The cluster formed with the CA algorithm contains 71 images and only 47 images of the good category (see figure 4). The cluster formed with the SOON algorithm has no noise but contains only 14 images (among 72) (figure 5). With our method, a cluster of 88 images is found, with 18 noisy images and 70 good images.

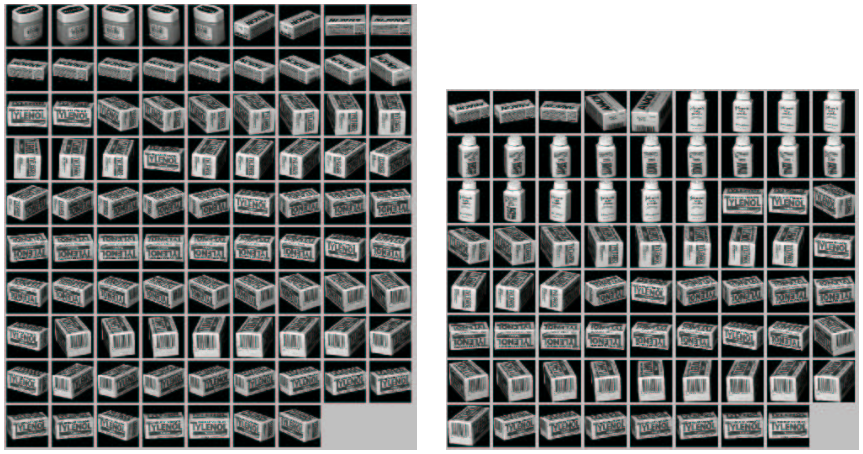

Fig. 4. left: cluster of object 'drugs package' obtained by ARC, and right: cluster of object 'drugs package' obtained by CA algorithm

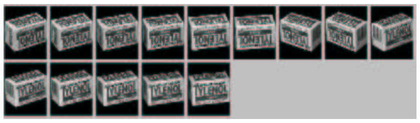

Fig. 5. cluster of object 'drugs package' obtained by SOON algorithm

The CA algorithm suffers from the noisy data which prevent it from finding the good clusters.

On the contrary, the SOON algorithm rejects lot of images in the noise cluster: thus good clusters are pure, but more than a quarter of the database is considered as noise. Since whole categories can be rejected (table 2 shows that 2 complete categories of Columbia database are in the noise cluster) the image database is not well represented.

ARC method avoids these drawbacks. It finds clusters which contain almost all images of the natural category, with a only small amount of noise. The noise cluster contains only really ambiguous images which would affect the results by biasing the clustering process.

5 Conclusion

We have presented a new unsupervised and adaptative clustering algorithm to categorize image databases: ARC. When prototypes of each category are picked and collected together it provides a summary for the image database. It allows to face problems raised by image database browsing and more specifically handle the “page zero”. It allows computing the optimal number of clusters in the dataset. It assigns outliers and ambiguous image signatures to a noise cluster, to prevent them from biasing the categorization process. Finally, it uses an appropriate distance to retrieve clusters of various shapes and densities.

References

1. Bezdek, J.C.: Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press (1981)
2. Gustafson, E.E., Kessel, W.C.: Fuzzy clustering with a fuzzy covariance matrix. In: IEEE CDC, San Diego, California (1979) 761–766
3. Dave, R.N.: Characterization and detection of noise in clustering. Pattern Recognition Letters **12** (1991) 657–664
4. Frigui, H., Krishnapuram, R.: Clustering by competitive agglomeration. Pattern Recognition **30** (1997) 1109–1119
5. Boujemaa, N.: On competitive unsupervized clustering. In: Proc. of ICPR’2000, Barcelona, Spain (2000)
6. Brunelli, R., Mich, O.: Image retrieval by examples. IEEE Transactions on Multimedia **2** (2000) 164–171
7. Medasani, S., Krishnapuram, R.: Categorization of image databases for efficient retrieval using robust mixture decomposition. In: Proc. of the IEEE Workshop on Content Based Access of Images and Video Libraries, Santa Barbara, California (1998) 50–54
8. Frigui, H., Boujemaa, N., Lim, S.A.: Unsupervised clustering and feature discrimination with application to image database categorization. In: NAFIPS, Vancouver, Canada (2001)
9. Nene, S.A., Nayar, S.K., Murase, H.: Columbia object image library (coil-20). Technical report, Department of Computer Science, Columbia University, http://www.cs.columbia.edu/CAVE/ (1996)
10. Niemann, H.: Pattern Analysis and Understanding. Springer, Heidelberg (1990)
11. Huang, J., Kumar, S.R., Mitra, M., Zu, W.J.: Spatial color indexing and applications. In: ICCV, Bombay, India (1998)
12. Yager, R.R.: On ordered weighted averaging aggregation operators in multicriteria decision making. Systems, Man and Cybernetics **18** (1988) 183–190