



HAL
open science

FReBIR : an image retrieval system based on fuzzy region matching

Sylvie Philipp-Foliguet, Julien Gony, Philippe-Henri Gosselin

► **To cite this version:**

Sylvie Philipp-Foliguet, Julien Gony, Philippe-Henri Gosselin. FReBIR : an image retrieval system based on fuzzy region matching. *Computer Vision and Image Understanding*, 2009, 113, pp.693-707. hal-00519447

HAL Id: hal-00519447

<https://hal.science/hal-00519447>

Submitted on 20 Sep 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

FReBIR : an image retrieval system based on fuzzy region matching

Sylvie Philipp-Foliguet, Julien Gony, Philippe-Henri Gosselin

Equipes Traitement des Images et du Signal,
CNRS/ENSEA/UCP,
6 av. du Ponceau, 95014 Cergy Cedex, France
philipp@ensea.fr

Abstract

In this paper we present a method of image indexing and retrieval which takes into account the relative positions of the regions within the image. Indexing is based on a segmentation of the image into fuzzy regions ; we propose an algorithm which produces a fuzzy segmentation. The image retrieval is based on inexact graph matching, taking into account both the similarity between regions and the spatial relation between them. We propose, on one hand a solution to reduce the combinatorial complexity of the graph matching, and on the other hand, a measure of similarity between graphs allowing the result images ranking. A relevance feedback process based on region classifiers allows then a good generalization to a large variety of the regions. The method is adapted to partial queries, aiming for example at retrieving images containing a specific type of object. Applications may be of two types, firstly an on-line search from a partial query, with a relevance feedback aiming at interactively leading the search, and secondly an off-line learning of categories from a set of examples of the object. The name of the system is FReBIR for Fuzzy Region-Based Image Retrieval.

Keywords : Image indexing, image retrieval, inexact graph matching, fuzzy segmentation, spatial relation, kernel function.

1. Introduction

Content-based image retrieval can be seen as a specific problem of pattern recognition, particularly when it concerns the retrieval of images containing a specific type of object or animal, or a person, etc. in a general database of photographs. In this case, the main

problem is that the searched object can take various shapes and scales according to the shooting.

The solutions usually proposed to solve this problem of partial query are to extract characteristic points from the image and to represent images by features describing a neighbourhood of these points [23]. If this approach can be interesting to retrieve rigid objects, it cannot be used to retrieve animals or persons, because the shapes are too variable and the characteristic points not robust enough. Moreover some of the characteristic points are placed on the edges of the objects, and thus the features also include a description of the background of the object [31]. On the contrary regions obtained by segmentation better fit the content of the image and they carry more information than points. The aim of this paper is to develop a region-based method to perform partial queries. Regions can be described by internal features, and by their relative positions, such a representation is robust to large variations of shapes and scales. The issue is to match sets, of regions, that may have different cardinality. The matching process is based on the feature vector characterizing each region and the spatial relations between regions.

We are thus confronted with a dual problem : first to define a representation of the image by regions and then to match the set of regions composing the query to a set of regions from another image of the database in an acceptable time.

To solve the first problem, with the second objective in mind, we must represent the image as a set of regions which roughly corresponds to the main colour parts of the image. Our objective is to simulate the human visual system which perceives coarse zones with their approximate colours, sizes and their spatial relative positions. Despite the fact that our visual system does not perform an accurate segmentation of the scene, the recognition of a landscape or a painting is instantaneous. Thus we propose to use a segmentation, different from the other systems [10,18,26,30,37,47,51], which builds fuzzy regions. The main advantage is to be able to segment any image, even in difficult cases, when there is no clear limit between some parts of the objects. The regions are modelled by fuzzy sets, their contours are coarse and they may overlap, however they represent the image. The coarseness of the representation will be compensated by the matching step.

Since we address the problem of partial query, the query is composed by a set of regions, adjacent or not. The need for retrieving more than one region arises when the semantic object (for instance a person) is split into two or more regions - which happens often, due to contrasting colours and textures (in the example face, hair and clothing) ; but also when the user is interested in a set of several objects possibly scattered in the image.

The second problem is to match subsets of regions, by taking into account both the intrinsic features of each region and the spatial relations between regions. The subsets may be of different sizes. More precisely the query is composed by a set of (fuzzy) regions. These regions are coarse but they are characterised by a feature vector and by their spatial relations. The problem is then to find similar sets of regions, from other images, regarding their feature vectors and their spatial disposition even if they do not have the same cardinality than the query set. We propose an algorithm able to find a solution to this problem in a time compatible with an on-line use. It uses a similarity measure between sets of regions and it is thus able to rank sets of regions and images with regard to the query.

Since the objects we try to retrieve can be very variable, the search cannot be efficient with a single example of the category. The system has to learn several examples and counter-examples. This will be done either interactively through a relevance feedback process or with an off-line learning.

Our contributions are a new representation of the image as a set of fuzzy regions with their features, and a matching between sets of regions, compatible with their features, their composition. We have developed a Fuzzy Region-Based Image Retrieval system (FReBIR) compatible with real-time use and dedicated to object retrieval.

In section 2, we will explain how we obtain a fuzzy segmentation of images, which constitutes a robust representation of the images. The image signature, characterising both the regions and their spatial relations is described in section 3. In section 4 we present our algorithm for region set matching, the problem being to avoid the combinatory explosion. Results on various cases are displayed in Section 5. An extension to learn from a set of examples, either through a relevance feedback or with a learning set is presented in section

6, with results on a database of several thousands of images. Finally section 7 concludes the paper and gives some perspectives.

2. Fuzzy segmentation

Segmentation is defined as a partition into regions which are crisp sets : each pixel of an image belongs to one and only one region. But this definition of the segmentation does not take into account the fact that a pixel in the “centre” of a homogeneous region belongs certainly much more to this region than pixels agglomerated afterward, during a region growing, for example. To model this kind of knowledge, we propose to build regions which are no more crisp sets – as in the classical segmentation – but fuzzy sets. The advantage of this representation is that each pixel of the image has a membership degree to each region.

The term “fuzzy segmentation” can be found in several papers, but it is never clearly defined. The main ways leading to fuzzy segmentation, as mentioned by Bezdek [8] are fuzzy thresholding, fuzzy pixel classification, fuzzy rules, or region growing.

Fuzzy thresholding aims at defining membership functions for regions, based on a set of thresholds [39][14].

Fuzzy classification of pixels is widely used for remote sensing images, either in a supervised way from examples of the classes [54], or in an unsupervised way where classes are built from the data. In this case, most methods employ the fuzzy c-means algorithm (FCM) [7]. All algorithms based on pixel classification do not lead properly speaking to segmentation, since they only classify the pixels into one or several classes but do not create contiguous regions. The number of classes is often a priori fixed. FCM is also used by Boujemaa *et al.* [9] to achieve a crisp segmentation on tomographic cardiac images.

Also in remote sensing domain, eCognition system [4] starts with a region merging process based on colour and shape criteria, which leads to a hierarchy of segmentations. After a fuzzyfication of the region features, a more accurate segmentation is obtained by applying fuzzy rules based on linguistic and fuzzy concepts of the remote sensing domain.

In the region growing methods, the problem is to find the seeds of the regions, the function linking region homogeneity, and the membership grades. Moghaddamzadeh *et al.*

[38] have developed a complex algorithm of region growing limited by edges. The algorithm consists of two steps. The first step performs a coarse segmentation and seeds are chosen following a double criterion: most frequent colours and distance from edges. In the second step, which leads to a fine segmentation, the seeds are the pixels far from any of the regions of the first step. The membership function takes into account the colour homogeneity and the distance to seeds. In [51], region growing is carried out by fuzzy rules involving fuzzy criteria such as region homogeneity, region size or gradient sharpness. Ma *et al.* [33] perform a fuzzy region growing on a saliency map. Seeds are the most salient points and the growing is achieved by a fuzzy 2-partition of pixels into attended and unattended areas. The result is a crisp segmentation.

Our aim was to develop an algorithm as general as possible, that is to say not dedicated to a particular type of image. Our algorithm is based on a region growing, the main idea is to link the membership degrees to a region with the distances to the seed of the region. But seeds are not a priori fixed nor determined in a first step like in [38]. They are obtained during the region growing.

The result is a set of coarse regions (fuzzy sets) with the following properties :

- uniformity in terms of colour;
- contained expansion by high gradient norms;
- uncertainty where two (or more) regions encounter.

2.1. Definition of the fuzzy segmentation

Let Ω be a finite referential (set of n pixels). A fuzzy region R_j is a fuzzy set of Ω defined by a mapping μ_j from Ω to $[0, 1]$.

Definition: A *fuzzy segmentation* of Ω is a set of m fuzzy regions R_j whose supports are included in Ω and defined by the two following axioms. If $\mu_j(s)$ is the membership degree of pixel s to region R_j , then:

1. $\forall s \in \Omega, \forall j \in [0, m], \mu_j(s) \in [0, 1]$
2. $\forall j \in [0, m] \ 0 < \sum_{s \in \Omega} \mu_j(s) < n.$

Membership degrees are between 0 and 1, hence they equal 1 for the pixels of the core and 0 for the pixels that do not belong to the fuzzy region. The second axiom means that a

fuzzy region must not be empty and must not be complete (equal to Ω). Regions may overlap each other.

This definition is based on Ruspini definition of a fuzzy partition [48], but without the third axiom that imposes normalisation : for each pixel, membership degrees to all regions should sum up to 1. This axiom is not desirable for segmentation as shown by Krishnapuram [26]. On one hand, in case of overlap, it would penalise a region core, just because it also belongs to another region. On the other hand, it would artificially increase the membership of pixels far away from all cores, and whose membership is low for any region, as soon as it belongs to only one region.

2.2. Algorithm of fuzzy segmentation

Our algorithm performs a cooperative contour / region approach [15]. In order to extract homogeneous regions, the algorithm is based on a region growing, and to constrain the regions by the edges, this growing is performed on the gradient norm image. Region growing starts in homogeneous colour areas, which are region seeds. Pixels with high gradient norms act like dams, which are hard to cross. They can be bypassed, which allows the overcoming of the impulse noise. Since we want the algorithm to be automatic, the regions are automatically initialised in the zones of low gradient norms. Membership degrees to regions are then computed by means of a “topographic distance”, that we defined in [42], and which takes into account both the spatial distance between pixels and the difference of gradients.

The algorithm is composed of three steps. The first one performs the watershed algorithm, leading to the catchment basins. The second one merges the catchment basins, which are often too numerous after the first step. The merged catchment basins will constitute the regions. The third step performs the fuzzyfication of the regions and computes the membership degrees of the pixels to the fuzzy regions.

The watershed algorithm [53] considers the image as a surface in a 3D space $(x,y, g(x,y))$, with (x,y) a pixel of Ω and $g(x,y)$ its value in \mathbb{R} (the colour gradient norm in our case). The aim of the algorithm is to split this surface in so-called catchment basins. This term comes

from the geography : a drop of water falling on the surface will roll until one of the bottoms of the surface. The set of all points leading to the same bottom is called a catchment basin. Each catchment basin is initialized by a seed (composed by one or several pixels, located in the bottom of the basin). The watershed algorithm simulates the flood of the image (the 3D surface) from the seeds. In the classical version of the algorithm, a dam is build when two basins encounter, and the set of dams constitutes the watershed, the basins being the regions.

In our version of the algorithm, pixels at the junction of two basins are not labelled as watershed, but integrated into one of the two basins. This leads to a big number of basins, which are merged in the second step, according to criteria of size and relative depth [3]. Fuzzy regions are created from these merged basins. The membership degree of a pixel to a region is computed using the topographic distance (distance along the 3D surface) from this pixel to the seed of the basins constituting the region.

Our contributions mainly concern the third step (fuzzyfication of the regions).

For the first two steps, we used a map of colour gradient norms obtained by Di Zenzo method [20], but other function can be used, such as a saliency map [33]. We have adapted the watershed algorithm in the following way : pixels are increasingly processed in the order of their values (gradient norms) and every local minimum of the gradient norm is a seed of a basin.

In order to be compatible with storage on one byte, the membership degrees are multiplied by 255 and thus take values between 0 and 255. They are initialised to 255 for the seeds, and to 0 for all other pixels. Seed pixels are stored in a queue Q .

Initialisation of membership degrees during flooding

```

For each basin  $B$ 
  for each pixel  $s \in B$ 
    if  $s \in \text{seed}(B)$   $\mu_B(s) \leftarrow 255$ , put  $s$  into  $Q_B$ 
    else  $\mu_B(s) \leftarrow 0$ 
  end for
End for

```

Basins are merged according to their areas and depths : a basin too small or not deep enough is absorbed by its neighbour. The basin which absorbs basin B is denoted

Absorb (B). Moreover, the difference between the bottom levels of the basins is applied as a penalty on membership degrees of the absorbed one.

The gradient norm of pixel s is denoted $g(s)$.

Catchment basin merging

```

For any pixel  $s \in \text{seed}(B)$ 
   $h_B \leftarrow g(s)$ 
End for
While it exists  $A$  such as  $A = \text{Absorb}(B)$ 
  for any  $c \in \text{seed}(A)$   $h_A \leftarrow g(c)$ 
   $\forall s \in \text{seed}(B)$   $\mu_B(s) \leftarrow 255 - |h_B - h_A|$ 
   $B \leftarrow A$ 
End while

```

Each set of merged basins gives a fuzzy region. The membership degree of a pixel is conversely proportional to the topographic distance, which is defined as the length of the shortest path connecting the pixel with the seed, along the surface constituted by the gradient norm in the 3D space [42]. They are maximal (255) for pixels belonging to basin seeds (perhaps minus the difference of basin bottom levels). They decrease as pixels are going away from seeds : 1 for each spatial step (in 4-connectivity or in 8-connectivity), and a value proportional to the difference between gradient norms, until they reach 0.

Membership degrees to region R

```

For each catchment basin  $B$  of  $R$ 
  while  $Q_B$  is not empty
    extract  $s$  of  $Q_B$ 
    for each pixel  $v$  neighbour of  $s$ 
       $\mu = \mu_B(s) - (\lambda \cdot |g(v) - g(s)| + 1)$ 
      if  $\mu > \mu_B(v)$  then  $\mu_B(v) \leftarrow \mu$ 
      put  $v$  into  $Q_B$ 
    end for
  end while
End for

```

Thus pixels belonging to areas of a homogeneous colour, and thereby of a small gradient norm have large membership degrees in the corresponding fuzzy region. These degrees slowly decrease according to the spatial distance to the seed and strongly decrease when meeting an edge, zone of a large gradient norm. Impulse noise is bypassed, because a shorter path is found “around” it.



An image of our general database

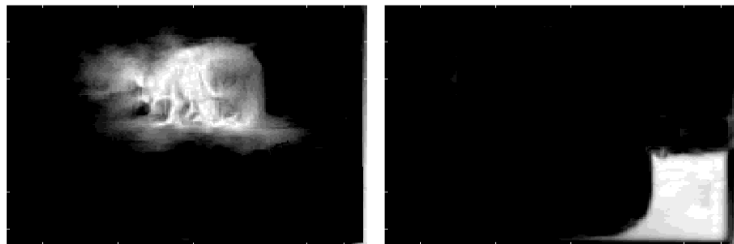


Fig. 1 : Two fuzzy regions obtained from the top image
left: region corresponding to the bear
right: region corresponding to a part of the water
(the lighter, the larger the membership degree)

Fig. 1 presents an example of fuzzy regions obtained by the algorithm. The boundary between water and bank, or water and bear shadow is very sharp. The corresponding fuzzy region sharply stops on these edges. On the contrary, the boundary of the bear sometimes mismatches with the grass, or with the ground. The fuzzy region spreads out more, its contours are less sharp. One can obtain a crisp segmentation by affecting every pixel to the region for which it has the largest membership degree. This "defuzzification" is only used to display simultaneously all the fuzzy regions. Notice that the fuzzy regions, more or less, overlap each other .

λ aims at balancing the spatial distance with the closest region seed and the difference of gradients. It will influence the spreading of regions between 0.5 for a large spreading and 3 for a small spreading. It has been set to 2 for all the tests and particularly for segmenting a whole database. The area threshold for basin merging induces the level of detail of the result. It is not beforehand set, it will increase or decrease from an initial value, until an expected number of regions. In the general database used in section 5, the number of regions must be between 10 and 30.

Hence the parameters are not crucial. They can be tuned for a complete database, according to a number of regions nearly fixed by the user.

3. Image signature

Each image of the database is indexed by a signature, which is composed of two parts: the first one is the set of features of all fuzzy regions, and the second one is a representation of the topology of these regions within the image.

Rosenfeld extended the definition of geometrical attributes (area, perimeter, etc.) to fuzzy sets [46]. The principle is to weight the contribution of each pixel by its membership degree to region R . The mass centre G of region R is defined by its coordinates:

$$x_G = \frac{\sum_{s \in R} \mu(s) \cdot x(s)}{\sum_{s \in R} \mu(s)} \quad y_G = \frac{\sum_{s \in R} \mu(s) \cdot y(s)}{\sum_{s \in R} \mu(s)} \quad (1)$$

where $x(s)$ and $y(s)$ are the spatial coordinates of pixel s .

The distribution of feature a for a fuzzy region R defined by its membership function μ was defined by Dubois and Jaulent [21]. The probability for any real r is defined by:

$$P_a(r) = \frac{\sum_{s \in R, a(s)=r} \mu(s)}{\sum_{s \in R} \mu(s)} \quad (2)$$

Distribution are thus obtained by adding the membership degrees of the pixels belonging to each class ; a normalisation is then performed. Thus pixels with small membership degrees – belonging to transitions or outliers inside a region – have little influence on the distribution shape. CIE $L^*a^*b^*$ space is used for colour, and twelve Gabor filters in 3 different scales and 4 orientations are used for texture analysis. Both spaces are quantified using an enhanced version of k -means algorithm [43].

The number of classes (or codewords) is very variable according to systems : in VisualSEEk [50], the HSV colour space is divided into 166 regular bins. Tests led in [17] with a number of classes varying from 8 to 400 showed that 25 is a good compromise between compactness of the signature and efficiency. However it is important that the codebook is adapted to the base [24].

Each image of the database is first quantified in 256 codewords for colour and 256 codewords for texture. Thus the 256 colour (resp. texture) codewords for each image are quantified in 25 colour (resp. texture) codewords.

In the following tests, the regions are described only by the colour and texture distribution (a vector of 50 values by region) and the coordinates of the mass centre. These coordinates are normalised by the dimensions of the image, thus taking values between 0 and 1.

There are various means to represent spatial relations between regions. Regions themselves can be represented by a point (generally the centre of mass), a rectangle (rectangle hull for example), etc. Spatial relation can be Boolean, such as “left to”, “above”. They usually involve the centres of mass or some projections on reference axes, like the 2D-strings [13], and their extensions 2-DC-strings, 2-D-B strings and so on [4], which more or less extend Allen's relations [2]. Malki *et al.* [34] also use Allen relations to define topological relations and orientation relations (such as “overlapping”, “covering”, “disjunction”) between objects. Egenhofer and Franzosa [22] have defined 9 topological spatial relations between 2D sets which can be extended to n-dimensional spaces. All these descriptors are binary.

Spatial relation can also be described by numeric values. Matsakis *et al.* [35] extended the notion of histograms of angles [27] and store in histograms of force information concerning the relative directional position between two objects. Fuzzy attributes of relative positions were proposed in [10] for crisp sets and in [27] for fuzzy sets. Berretti *et al.* defined “weighted walkthroughs” from relative positions in 9 directions [6]. All these features aiming at describing relative positions of sets are computed from relative positions of pixels. Thus they have the drawback to be time-consuming, and, according to our tests, without considerable improvement of results. Numerous recent works address this problem of representing spatial relations between objects, and a comparison between the various methods in the framework of image retrieval would be interesting.

In this paper, we do not focus on the choice of the most suited representation. Methods aforementioned are often too precise to retrieve a set of regions having very variable shape from an image to the other. Furthermore they are not necessarily suited to fuzzy regions. We propose a solution to match sets of regions nearly located in the same spatial disposition. For instance, in the case of a car (viewed by its side), wheels are situated under the

bodywork, windows are at the top and all this is generally over a dark zone (shadow and tar).

Tests in which we had allocated precise values to edges such as fuzzy relations [10] or more simply distances between mass centres showed us that this approach is not relevant for our problem, because of the high variability of the researched objects. The simplest way to represent the adjacency is the adjacency matrix of the regions with a value 1 if both regions have at least one pixel in common, otherwise 0. A more precise solution would be to store the degree of overlapping of regions.

The signature of each image is thus made up of two parts, on one hand information about regions : area, colour and texture distributions, coordinates of the mass centre and on the other hand the adjacency matrix of regions.

The relative position of regions is taken into account during the match process through the relative positions of their mass centres.

4. Matching algorithm

We are interested in a query constituted by a set of regions, adjacent or not. The problem is to retrieve images represented by sets of regions which matches at best this query.

After segmenting into regions, each image of the database is represented by an attributed relational graph (ARG). Nodes correspond to regions and edges to adjacencies between regions. Each region is characterised by a set of features (colour, texture, ...) and edges are characterised by spatial information such as "above", "on the left of", overlap, etc.

The query is a sub-graph of the ARG representing the query image. It may be made up of one or several connected components. The search consists then in looking for the sub-graph which matches the best the query ARG (cf. Fig. 3), in each image of the database.

4.1. Related works

There are various approaches to solve the problem of image retrieval based on region matching.

Image composition is sometimes simply not taken into account as in [30] or in Blobword [11] and the final score for a set of regions is a simple operator such as the *min*, the *max*, or the *mean* of the scores of the one-to-one region match. Spatial information can be simply coded through the coordinates of the mass centre. It is used in [28] as a feature as well as a colour feature during the segmentation process. In [1] the image is represented by a vector coding on one hand blocks belonging to a given patch vocabulary and on the other hand their spatial relations coded on 20 possible relations (5 distances \times 4 directions). In NeTra [32], the spatial localisation of a region is measured by two rectangles (inner and outer), which are used to constraint and reduce the retrieval. Explicit relation between regions, such as adjacency is not considered. In [44] the image is split into 9 blocks of same size, and the only label of the dominant region in every block is stored. The spatial structure is implicitly coded, and images are retrieved very quickly, by the way of a hash structure. In VisualSEEK [50], the query does not consist of image regions, but of a set of coloured patterns. For each pattern, a list of candidate regions is built, and the combinatory is further reduced by relative and absolute relations between regions, compared to those of the patterns.

The most general systems allow the match of two sets of regions (respectively belonging to the query and to the target image), with a possible different number of regions.

In Picasso system [19], the retrieval of a region set is performed by a recursive algorithm which matches one query region with a node of the multiresolution pyramid of segmented images. A region at a given resolution is the union of the regions at finer resolutions, and a query region can be matched to a region of any resolution, so the number of regions in the query and in the target can be different. The distance between two regions takes into account the distance between the mass centre, and the similarity between images is the sum of the similarities between matched regions. In SIMPLIcity [55], the query is a set of regions, and the distance between two sets is the sum of the distances between matched regions, weighted by a score of significance of the match. The distance does not take into account spatial information, but the method allows the matching of one region of an image to several regions of the other. The algorithm matches at first the pairs of regions the most

similar and weights the distance by a “significance credit”, function of the area of the regions. No relation between regions is taken into account.

The graph matching was used for a long time in pattern recognition. Relaxation [47] reduces a labelling (discrete, fuzzy or probabilistic) by considering the labels of the neighbour nodes. Hong and Huang [29] built a pattern ARG from a set of sample ARGs ; it is well adapted to logos or generally to objects submitted only to rigid deformations. Robles-Kelly and Hancock [45] propose an algorithm of inexact matching which only takes into account the node adjacencies, the nodes themselves are not attributed. Berretti *et al.* [4] use the A*algorithm to perform ARG isomorphism.

Medasani and Krishnapuram [36] make an inexact matching of fuzzy graphs, in which regions are affected with symbolic labels. The relative positions of regions are taken into account through Bloch [10] fuzzy relations. In order to reduce the computing time for the matching, graphs are prior clusterized into graph clusters represented by a prototype. Only the graphs belonging to the cluster the closest to the query are compared. Perching and Bloch [41] defined a general formalism of fuzzy morphism between graphs. Our algorithm belongs to this framework because it defines a mapping on the pairs of matched nodes (a distance between regions) and a mapping on the pairs of matched edges (binary values indicating whether these edges are in the same relative positions in both the query image and the target image).

The matching algorithm presented in this paper is of the inexact type since node attributes may differ. Our algorithm is not limited to graph isomorphism, since one node of a graph may be matched to several nodes of the other graph. Two matched subgraphs have not to be identical in terms of node number, node attributes and edge number but only similar for node attributes (measured by an appropriate similarity measure expounded below) and consistent for the edges. Since we are interested in spatial relations between regions, the relations stored in the ARG concern the adjacency. The relative positions of regions are taken into account during the matching in order to reduce the combinatorial. The problem we address is NP-complete, but the solution has to be found in an acceptable time for a user making a search in the database.

Cordella *et al.* presented an algorithm of inexact matching of ARGs [18] : two graphs are similar if they can be made isomorphic to each other by a set of transformations. These transformations are the split of a node into a sub-graph, the merge of a sub-graph into a node and the insertion or the removal of a branch.

To solve this NP-complete problem in a reasonable time, we use a tree structure inspired from [18]. The advantage of such a structure is that with appropriate heuristics, all possibilities of matches of region pairs have not to be explored. Search is concentrated on branches which are susceptible to lead to the solution. Heuristics concerns the order of examination of the region pairs, and takes into account distances between regions and topological consistency. Moreover it can be extended to the comparison between a graph and a set of graphs in a classification task.

4.2. Graph matching applied to image retrieval: search tree

As Cordella *et al.* [18], we used a search tree (Fig. 2) and we propose a solution for a fast construction of this tree adapted to the problem of region set matching. Our purpose is to obtain an optimal (or sub-optimal) solution according to a given similarity function.

The query regions are denoted $R_i, i = 1, \dots, n$ and the set of regions of a target image are denoted $S_j, j = 1, \dots, m$

For each image of the database (called target image) the distance (see section 4.4) between all pairs of query/target regions are computed. At level i of the tree (see Fig. 2), a node represents a match between query region R_i and one region of the target image S_j .

Edges between nodes represent topological consistency between pairs of regions. For example, the consistency between (R_i, S_j) and (R_{i+1}, S_k) can take into account :

- the adjacency of (R_i, R_{i+1}) and (S_j, S_k) which must be of the same type (adjacent or not);
- the relative position of (R_i, R_{i+1}) and (S_j, S_k) , for example if R_i is above R_{i+1} , S_j must be above S_k .

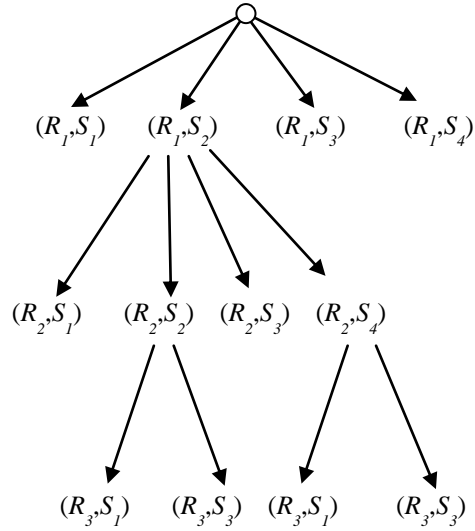


Fig. 2 : Search tree: each node corresponds to a match of two regions (R_i : query region, S_j : candidate region), each arrow corresponds to a topology compatibility

A match between two sub-graphs corresponds to a path from the root to a leaf. The dissimilarity between sub-graphs is the weighted sum of the dissimilarities for all nodes of a path :

$$\frac{1}{m} \sum_{i=1}^m w_i d(R_i, S^i) \quad (3)$$

where S^i is the target region matched with R_i , d represents the dissimilarity between region features (see section 3), w_i is a non negative weight and m is the number of query regions.

As dissimilarities and weights are positive or null, each new node added to a path increases the total dissimilarity of this path and never decreases it.

The tree building and its pruning are performed during the search and has to be fast enough to be compatible with a “real-time” use. The tree has at most $m \times n$ nodes if the target image is composed of n regions.

A classical heuristics for this “branch and bound” problem is to try the most promising solution at first. The tree is generated by a depth-first search procedure. For each node, the son which has the best chance to lead to the path of minimal dissimilarity is examined at first. That is to say for each node of level i , the nodes of level $i+1$ are examined in the order

of increasing dissimilarity. Thus the first path of length m is built by constructing at each node the branch leading to the pair of minimal dissimilarity. This path is not necessarily the best path, but it gives a first solution for the graph matching. The depth-first procedure will then look for better solutions compared to this initial one.

By this way, the optimal solution is always found, without explicitly building all the possible paths. However this exhaustive search can be too slow (for too many query regions for example), the search can then be stopped at any time, (usually after a fixed amount of time) the found solution is then sub-optimal.

The properties of this tree are :

- there are as many levels in the tree as query regions ;
- as soon as the global dissimilarity of a path exceeds the value of the most promising one, the current node is not developed ;
- the optimal solution is always found ;
- a sub-optimal solution can be found in a given amount of time.

4.3. To complete the matching

The drawback of the use of a search tree is that one region of the target image can match several query regions, but not the contrary. This allows us to manage an under-segmentation of the target image (Fig. 4a) with regard to the query image, but not an over-segmentation (Fig. 4b). A solution is to systematically over-segment the query image, that is what will be done when using a learning set for image category search (section 6.2). Another way is to add a second step in the following way. The neighbours of every target region are checked in order to retrieve regions likely to match the query region. More precisely, for each match (R_i, S_j) , all neighbours S_k of S_j are examined. If $d(R_i, S_j \cup S_k)$ is lower than $d(R_i, S_j)$ then R_i is also matched to S_k . In the example of Fig. 3, R_3 matches S_1 , S_1 and S_3 are adjacent, the match R_3 with $S_3 \cup S_1$ improves the match (R_3, S_1) , so R_3 is also matched to S_3 .

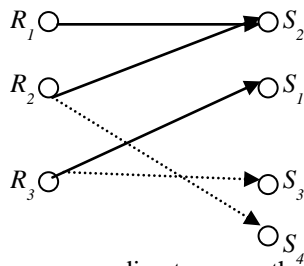
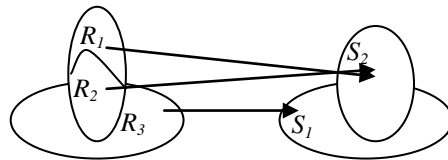
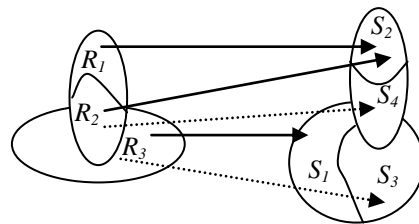


Fig. 3 : Initial matching, corresponding to one path of Fig. 2 (continuous arrows), completed by the second step (dot arrows)

A match between two ARGs is then represented by a set of pairs of similar query region - target region whose adjacencies and relative positions are the same in both sub-graphs.



a. Example of an under-segmented target image, corresponding to the initial match of Fig. 3



b. Example of an over-segmented target image corresponding to the complete match of Fig. 3 (continuous arrows: first step, dot arrows: second step)

Fig. 4 : Two examples of inexact graph matches with different region numbers

4.4. Dissimilarity between subgraphs and between images using kernel functions

We will see in the next section that in order to retrieve images containing a category of objects, these objects being very variable in shape and colour, we need several examples of the category. Thus we need a dissimilarity function not only between two regions (target and query) but also between one region (target) and a set of query regions.

The retrieval of a category of images is a two-class classification problem. As there is no linear solution in \mathbb{R}^k (if k is the dimension of the representation space), the data are mapped into a higher-dimensional space H , where a linear separation between classes can be found. Let's note Φ the mapping, $\Phi: \mathbb{R}^k \rightarrow H$. H is an Hilbert space, that is to say affected with a dot product denoted $\langle \cdot, \cdot \rangle$. H can be of huge dimension (and even infinite), so instead of working directly in H , we use the "kernel trick", which consists in computing a dot product in H : $K(x, y) = \langle \Phi(x), \Phi(y) \rangle$. K is a kernel function (positive and semi-definite)

Thus the distance between two regions R and S respectively represented by the normalised vectors ($R^c, c=1, \dots, k$) and ($S^c, c=1, \dots, k$) is defined by :

$$d(R, S) = \left\| \Phi(R^c) - \Phi(S^c) \right\|_2 \quad \text{with} \quad \|u\|_2 = \sqrt{\langle u, u \rangle} \quad (4)$$

A dissimilarity function f is associated to each root-leaf path of the tree, which corresponds to an initial match of subgraphs. This dissimilarity function is calculated during the tree building and used for the pruning.

Hence an initial match M corresponds to a path in the search tree composed of m nodes, if m is the number of query regions. M is a set of m matched pairs query region – target region:

$M = \{ (R_1, S^1), (R_2, S^2), \dots, (R_m, S^m) \}$, where R_i is a query region, and S^i the matched target region.

Considering that all the regions have the same importance, the dissimilarity measure is a weighted mean of the dissimilarities of all matched regions. The weight w_i aims at measuring the consistency of two pairs of matched regions $(R_i, S^i), (R_{i-1}, S^{i-1})$. Hence the dissimilarity between matched subgraphs represented by M is :

$$f(M) = \frac{1}{m} \sum_{i=1}^m w_i d(R_i, S^i) \quad (5)$$

The second part of the algorithm adds region matches. When match (R_i, S^j) is added (cf. section 4.3) to match (R_i, S^i) , $w_i d(R_i, S^j)$ is added to $f(M)$ and all weights are renormalized by $1/(m+1)$.

Finally, the dissimilarity measure for each target image is the minimum of the dissimilarities over all possible matches of the target image with the sub-graph constituting the query.

5. Results

In the following tests, we only used the vertical consistency. We think that for general databases, composed of landscapes, persons, animals, etc., only the vertical position is of importance to retrieve similar images. Symmetry over a vertical axis does not change our perception of the image, while symmetry over a horizontal axis changes it a lot. All weights w_i are set to 1.

We considered several databases, all images are automatically segmented by our algorithm of fuzzy segmentation and signatures are computed as explained in section 3. We used a histogram of 25 colors and 25 textures for each image, computed from a vector quantization [17]. For the first two databases, we have “ground truths” which give the category of each image.

The first database is a general database of 1200 images of photographs, texture and so on. It contains 25 categories, such as bear, elephant, car, texture and so on. We built new ones in order to perform the evaluation, such as “red car”.

The CBIR system performance measurement is based on the Precision and Recall :

$$\text{Precision} = \frac{\text{number of relevant retrieved images}}{\text{number of retrieved images}}$$

$$\text{Recall} = \frac{\text{number of relevant retrieved images}}{\text{number of relevant images in the category}}$$

Precision and Recall are considered for one category, and have as many values as images in the database (from one image retrieved to the maximum number of images the system can return). Precision and Recall are metrics to evaluate the ranking of the images returned by the system for one category. The Precision curve is always decreasing (or stationary), and the best Precision curve is the one which decreases the less, which means that whatever the number of images retrieved by the system, most of them are relevant. The Recall curve is always increasing, and the best Recall curve is the one which increases the fastest, which

means that the system has retrieved most of the relevant images, and few of them are lost. Precision and Recall are interesting for a final evaluation of one category, however for larger evaluation purposes, we consider the Precision/Recall curve. This curve is the set of all the couples (Precision, Recall) for each number of images returned by the system. The curve always starts from the top left (1,0) and ends in the bottom right (0,1). Between these two points, the curve decreases regularly. A good Precision/Recall curve is a curve which decreases slowly, which means that at the same time, the system returns a lot of relevant images and few of them are lost. This property is interesting since the size of the category is not playing an important role, which allows the comparison of Precision/Recall for different categories. The Precision/Recall curve can also be summarized by a single real value called Average Precision, which corresponds to the area under an ideal (non-interpolated) recall/precision curve.

To evaluate a system over all the categories, the Average Precisions for each category are combined (averaged) across all categories to create the non-interpolated Mean Average Precision (MAP) for that set. Let's note that this criterion is the one used by the TRECVID evaluation campaign¹.

The Precision and Recall values are measured by simulating retrieval scenario. For each simulation, an image category is randomly chosen. Then 100 images are selected using active learning and labelled according to the chosen category. These labelled images are used to train a classifier, which returns a ranking of the database. The Average Precision is then computed using the ranking. These simulations are repeated 1000 times, and all values are averaged to get the Mean Average Precision. These simulations are repeated ten times to get the mean and the standard deviation of the MAP.

In Fig. 5, the problem is to retrieve a brown bear near water. The query is made of two regions covering the bear and a part of the water (these regions are those of Fig. 1). Fig. 5 displays the ranked retrieved images. Among the 16 first images retrieved by the system, 11 images represent a brown bear near water. There are some mismatches with other images, for example between water and sky, because the region features are very simple (just the

colour and texture distribution), and we used no relevance feedback. In Fig. 6 are displayed the first two retrieved images and, beside each of them, the fuzzy regions which match at best the two query regions. In both images, the retrieved regions correspond to bear and water, even if segmentations are quite different. Matched regions have shape different from the query regions, their size may differ a lot from the query regions and they are not at the same position in both images (the only constraint is that bear regions must be lower than water regions). With this example, one can see the interest of the fuzzy segmentation, which allows bigger flexibility in the shape of regions. It overcomes the imperfection of the segmentation, since it allows retrieving images (or objects) even if the segmentation is quite different, over-segmentation or under-segmentation for one of the images.

Another example displayed Fig. 7 shows that red cars can be retrieved from a query composed of a part of the bodywork and a wheel. Among the 25 closest images, 24 include a red car, in various positions. Fig. 8 displays the precision/recall curve for this query.

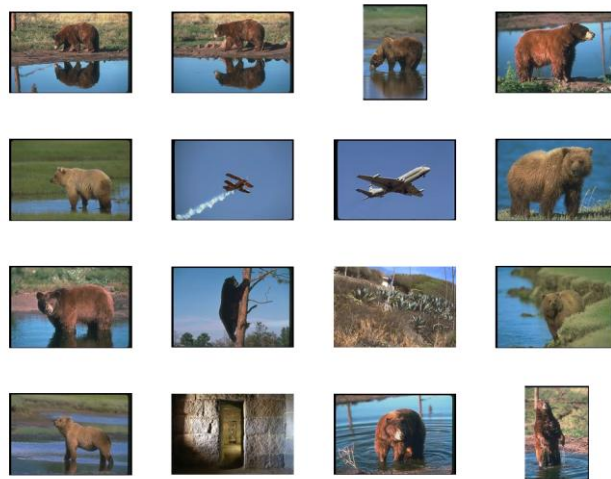


Fig. 5 : Retrieval result from a query made of two regions (the bear and a part of the water displayed Fig. 1). 11 correct images out of 16.

¹ <http://www-nlpir.nist.gov/projects/trecvid/>

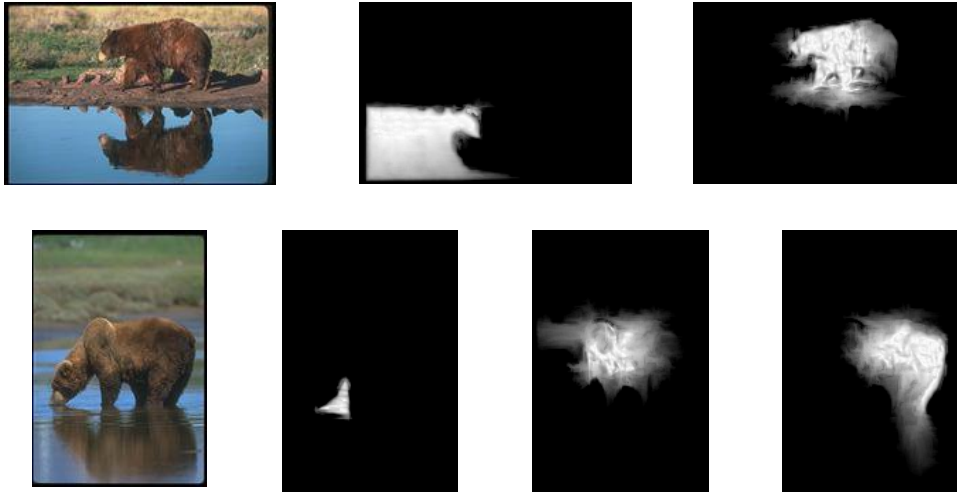


Fig. 6 : The first two images retrieved and for each of them, the regions which best match the two query regions.

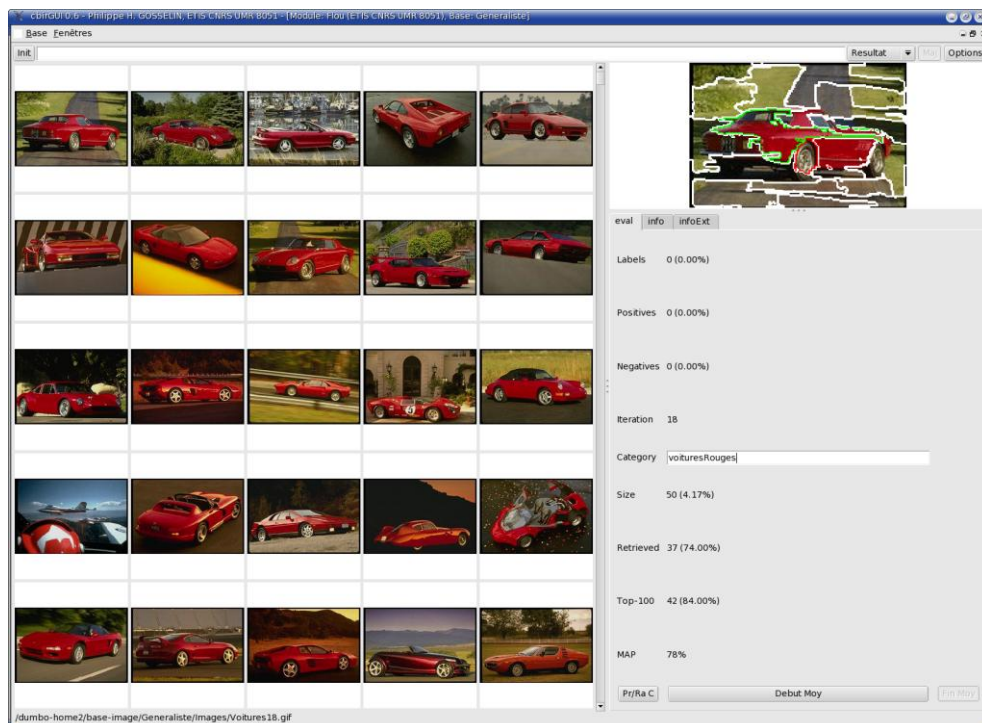
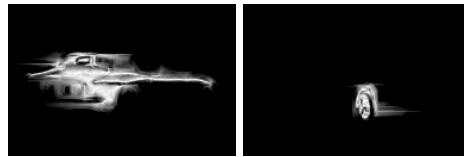


Fig. 7 : Search of red cars from a query made of the two regions displayed on top row. Among the 25 closest images, 24 include a red car.



Fig. 8 : Precision/recall curve for the "red car" category.

Then we have compared these results with a global query using a SVM classifier [17]. The signature for the SVM is the colorimetric distribution in the same colour classes than for the regions. The classifier starts with one image of the category, in order to work in the same conditions as the region-based system. Curves of Fig. 9 show results of the "red car" category, averaged out 50 researches, with 1, 2 or 3 regions covering a red car belonging to different images. With this example, one can conclude that if we have only one example of the category, our system is able to retrieve more images than a global classifier like SVM.

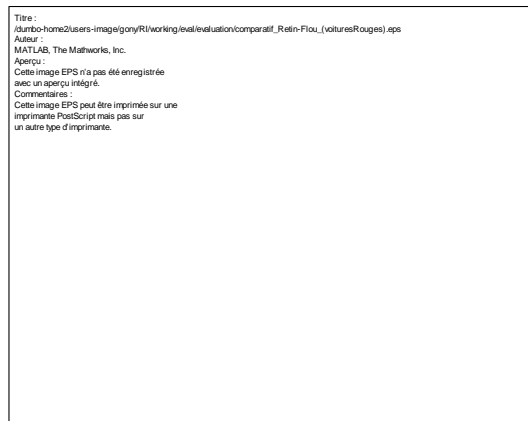


Fig. 9 : Comparison partial query / global query for the "red car" category.
 Average on 50 researches

Then we have studied the link between the number of query regions and the results. We have compared the retrieval of elephant images, from a query made of one, two or three regions and we have averaged on 50 requests of each type (on different images of elephant category). An example of query with two regions is displayed in Fig. 11. This comparison

(Fig. 10) as well as other similar tests on other categories show that it is of best interest to cover as much as possible the requested object. The redundancy induced by the overlap between regions is better than incomplete information included in a single region not covering the whole object.

This pleads in favour of a fuzzy segmentation which handles the imprecision of the contours of the objects in an image as well as those inside the objects. The colour of all pixels composing the object (in this example the skin, the tusk, the eye etc.) are taken into account in the signature.

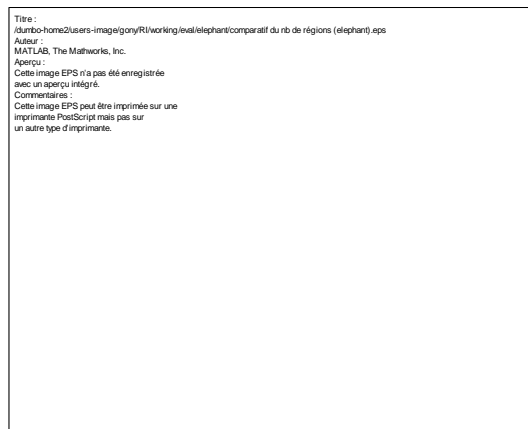


Fig. 10 : Comparison of queries made of one, two or three regions.
Average on 50 researches of "elephant" category.

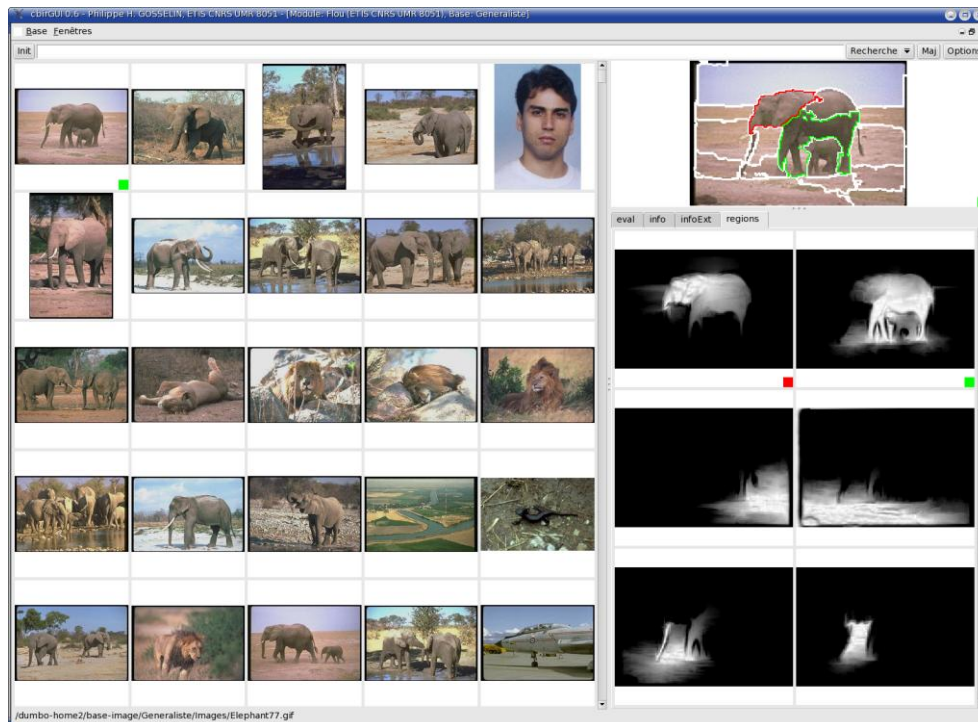


Fig. 11 : Result from a query made of two fuzzy regions covering the elephant : upper region is light grey, lower region is dark grey. On the right, the query image with the contours of the defuzzified segmented image. Edges are coloured in red and green. Below some of the fuzzy regions, first row the two query regions. On the left, the ranked result images.

6. Learning graphs of regions

If first retrieved results often include images of the searched category, the system is unable to manage the large variability of appearance of animals, persons or non rigid objects. Whatever the features used to represent a region, a simple system which works with a single query is condemned to remain in a local neighbourhood of the query. Only the user is able to conduct the search towards all modalities of the same semantic entity, unless a sufficient amount of examples are learned by the system. We have explored these two ways of learning an object category :

- start from a single partial query (a set of regions belonging to a single image) and perform a relevance feedback through interaction with the user
- learning from a set of examples and counter-examples

Both ways use the same process of machine learning, based on a statistical classifier, working with regions. Actually the system uses one classifier for each query region ; it is

trained from varied examples. For instance, if the searched category is “car”, a classifier is trained with regions of “bodywork”, and another classifier with regions of “wheel”. Any two-class classifier can be used, one class corresponds to one of the regions (for example, the class “bodywork”), and the other class corresponds to the rest of the regions of all images. The matching process works with the same search tree as in section 4.2. The only difference lies in the dissimilarity measure which no more involves a query region and a target region as in Eq (3), but here involves a target region and a “semantic” class of regions (for example “bodywork” or “wheel”). All examples of graphs for an object category must also share the same relative positions for the regions (adjacency, above, left to). In the example of “car”, regions “bodywork” and “wheel” must be adjacent and the bodywork must be above the wheel.

6.1. Interactive learning

We built a user friendly interface in which the user can choose the query image, the type of spatial relationships (vertical/horizontal/both/none), the classifier and the kernel function. The regions of the returned images are displayed with edges of the same colour as the corresponding query region (cf. Fig. 11). Hence, if all returned regions (those with coloured edges) correspond to the query region and respect the spatial disposition, the user annotates the image as relevant and all returned regions are considered as positive examples. On the contrary if no region corresponds to a query region, the image is annotated as irrelevant and all its regions are negative examples for all classes.

The feedback loop performs a two-class classification per query region, using all negative examples taken from all irrelevant images and the only positive examples of the class of the query region. The classification is updated at each iteration with the new annotated images, which gives new positive and negative examples. For each class, there are much more negative examples than positive ones, which properly models the asymmetry between the class representing the object (or a part) and the rest of the regions.

After each step of annotation, the graph matching is performed, by the way of a tree for each image of the database. The dissimilarity computed for each node of the tree now equals the distance between the region and the class of the query region.

Any two-class classifier can be used. We tried Bayesian [52], k -nearest neighbours, Fisher Discriminant [35] and SVM [12], all of them with various kernels functions [16].

At the beginning, there is only one graph query, so each classifier starts with only one example of the class (one region for each classifier). The classifiers work as one-class classifiers, and use the Euclidean distance of Eq (4) between regions. During the feedback step, each classifier has positive and negative examples of its class. If there are only positive examples (the user has only annotated relevant images), the classifier continues to work as a one-class classifier. But if the user has annotated both positive and negative examples (which is the most probable and the most efficient), each classifier works as a two-class classifier.

We have compared four classifiers and various kernel functions with the general database. A result of precision/recall curves, computed on all categories (Fig. 12) shows that the best classifier is the SVM with a Gaussian kernel with χ^2 distance :

$$\exp\left(-\frac{1}{\sigma}\sum_i\frac{(x_i-y_i)^2}{x_i+y_i}\right).$$

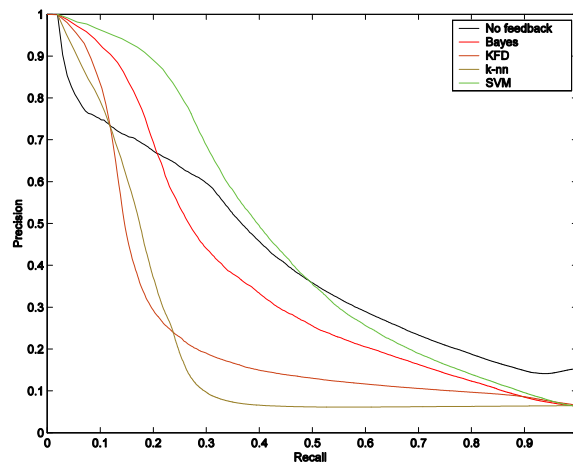


Fig. 12 : Precision/recall curves for 4 classifiers after 30 annotations (average on all categories)

Then we have compared a global approach (RETIN) [25] and our partial query method (FReBIR = Fuzzy Region-Based Image Retrieval) with the SVM classifiers. Signatures are computed with the same features, but they are computed on the whole image for RETIN (Fig. 13). First one can see the improvement induced by the iterations of relevance feedback

for FReBIR. Secondly the comparison between both approaches clearly indicates that, to retrieve images containing a specific object or animal (here white bears), a partial query provides better results (which is not true when semantics is defined in the whole image).

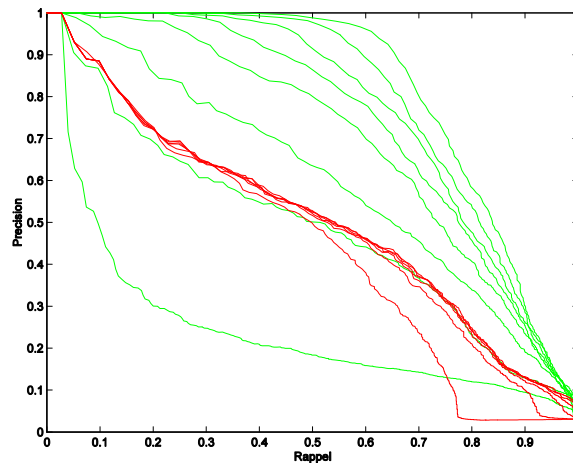


Fig. 13 : Precision/recall curves with 8 iterations of relevance feedback, for white bear retrieval ; in red RETIN, in green FReBIR

6.2. Learning with a learning set

The results presented in this section are part of a challenge ran in 2006 (ImagEVAL¹). The database used for this task is composed of 3 000 images of photographs of any types, colour and monochromatic.

One of the competition tasks is a task of object retrieval; more precisely it consists on retrieving images containing a given object. Ten different objects have to be found in a first challenge, such as “car”, “cow”, “US flag”, “Eiffel tower”. The training sets are composed of 20 images (for Eiffel tower) until 100 images (for car). The images of the training set contain one object of the category, with a background (see Fig. 14, Fig. 16, Fig. 18, and Fig. 20).

The images of the training set have dimensions included between 300 and 800, whereas images of the test set are larger, with dimensions between 600 and 1024. The fuzzy segmentation is performed in order to have between 8 and 14 regions for the images of the training set and between 15 and 20 regions for the images of the test set. The aim is to have

¹ <http://www.imageval.org/>

a light under-segmentation of the training set compared to the test set in order to avoid the second step of the matching.

The regions used for the training of each graph are manually chosen through the interface described in section 5 (cf. Fig. 7). The region number and the type of spatial relation are chosen according to the category. For “car” category, the graph is made of 3 adjacent regions, with a vertical consistency : bodywork with window on the top, bodywork in the middle and wheel with shadow at the bottom. For “cow” category, the graph is made of 2 adjacent regions, with no spatial consistency. For “US flag” and “Eiffel tower” categories, the graph is made of 2 adjacent regions, with a vertical consistency.

No image of the test set belongs to the training test. Although the great variability of the database (colour and grey levels images, old car and race ones, night and day photographs of Eiffel tower, ...), the images returned as the most likely to belong to the category (see Figs. 15, 17, 19, 21) are for the majority of them images of the search category. These objects have very various shapes (cf. US flag) and sizes (cf. cows).



Fig. 14 : An example for the category “car” : 2 query regions surrounded with blue (bodywork) and red (wheels and shadow) edges

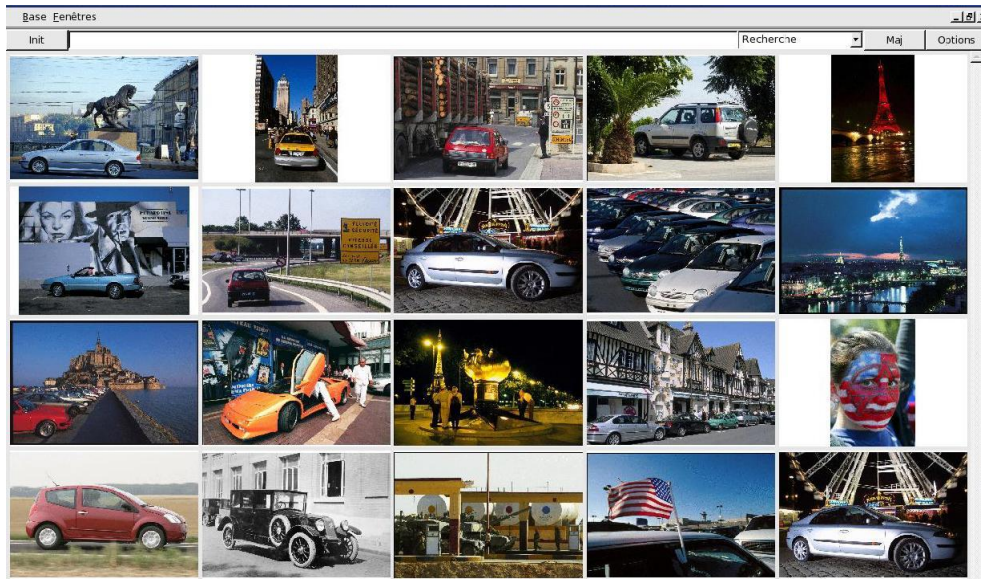


Fig. 15 : the 20 most relevant images retrieved for category "car" (15 correct)



Fig. 16 : An example for the category “cow” : 2 query regions surrounded with red (head) and green (black spot on the shoulder)

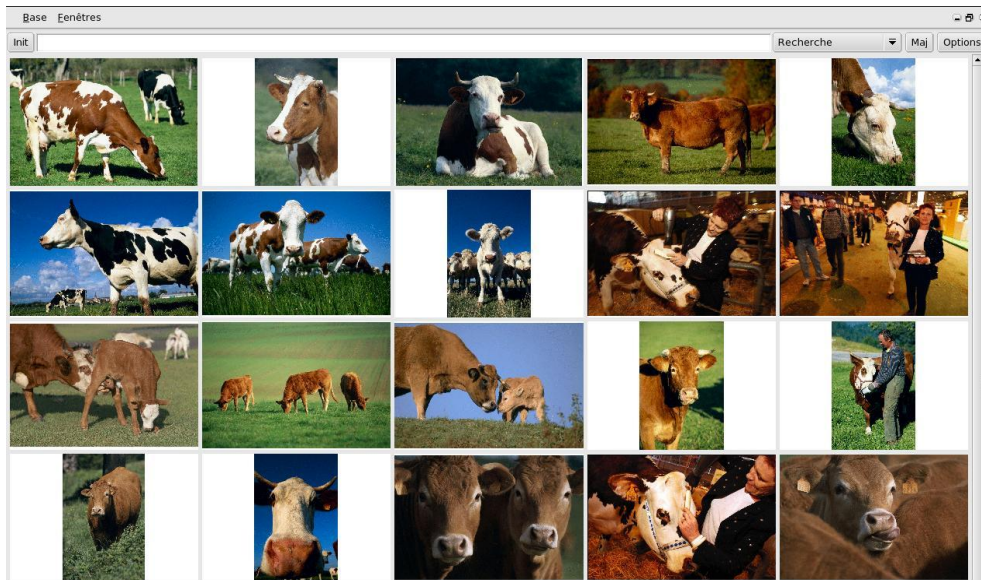


Fig. 17 : the 20 most relevant images retrieved for category “cow” (20 correct)



Fig. 18 : An example for the category “US flag” : 2 adjacent regions : stars (surrounded with red) and white and red strings(surrounded with green)

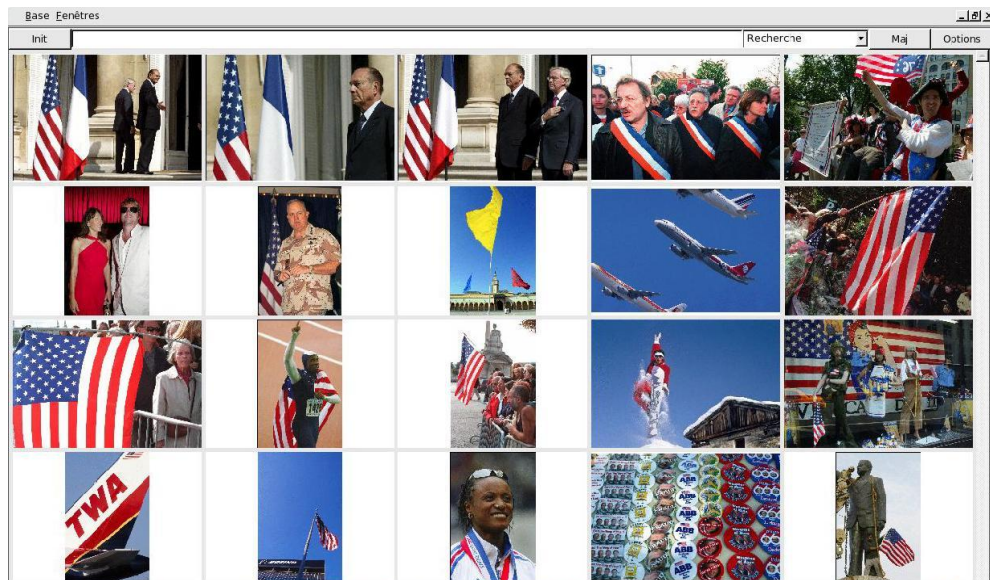


Fig. 19 : the 20 most relevant images retrieved for category “US flag” (12 correct)

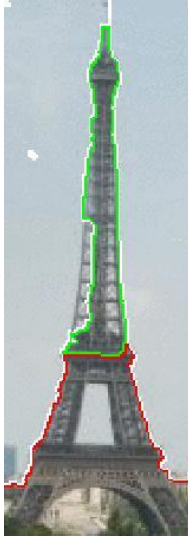


Fig. 20 : An example for the category “Eiffel tower” : 2 adjacent regions, one over the other, surrounded with green and red

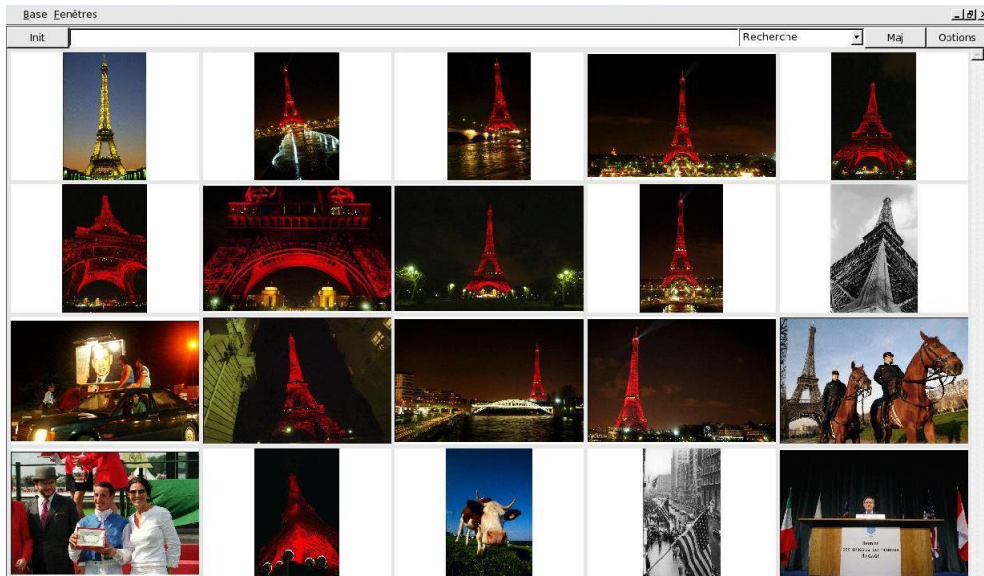


Fig. 21 : the 20 most relevant images retrieved for category “Eiffel tower” (15 correct)

We also tried our system on the Trecvid2006 data set. It contains 74523 keyframes, but as they represent videos, many of them are almost identical. Any system is able to retrieve these very similar images. Thus we used only a tenth of the dataset, and we tried to retrieve women portrait from the database. The query graph was composed of two regions. Figs. 22 and 23 display the results obtained after 30 annotations (16 positive and 14 negative) .

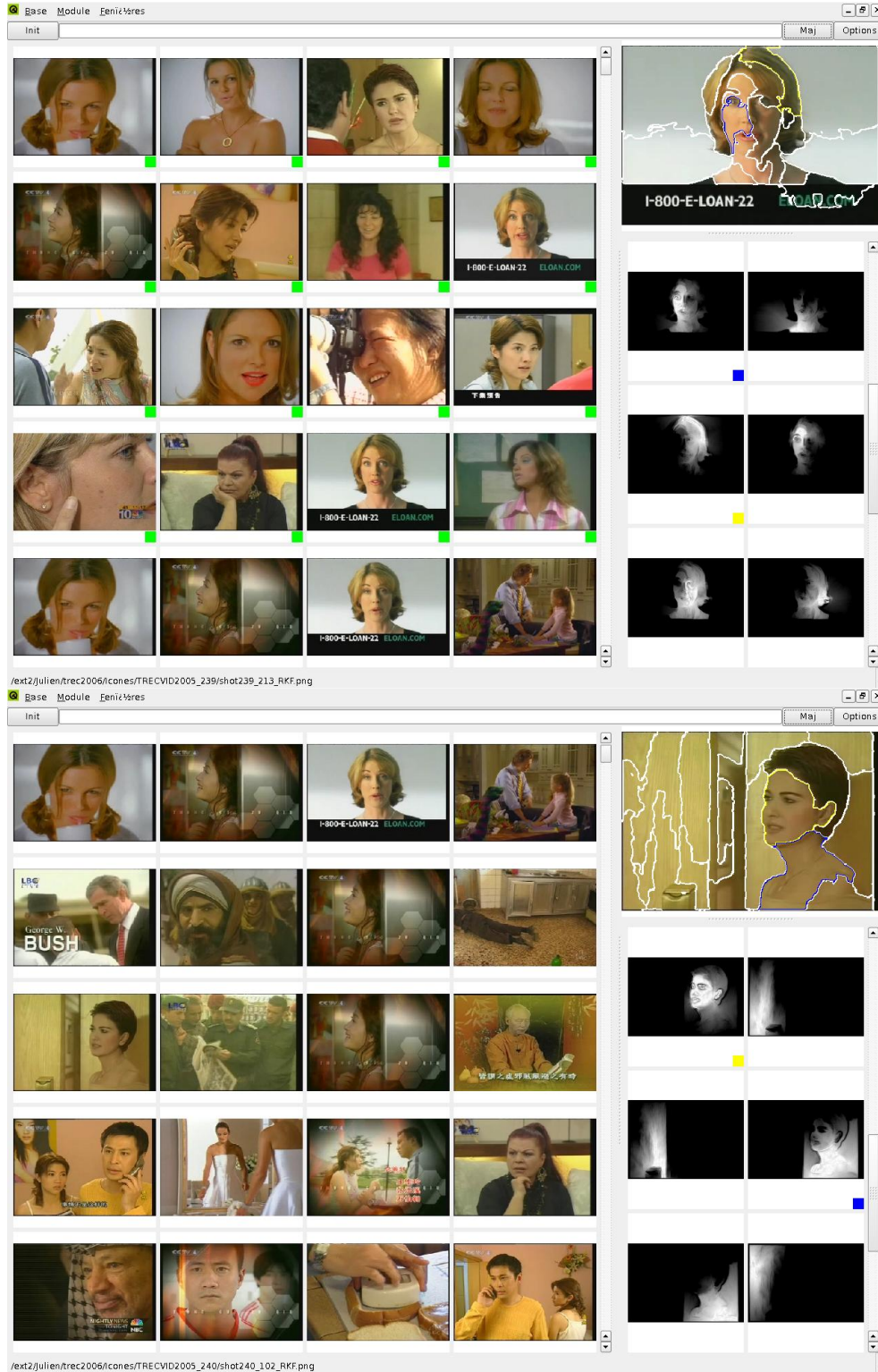


Fig. 22 : The 40 first retrieved images of woman portrait after 30 annotations (16 positive and 14 negative) of one tenth of the Trecvid keyframe database. On the left part, two images with some of their fuzzy regions. On the top, one query image, with the two query regions marked with blue and yellow squares, on bottom, one retrieved image, with the matched regions also marked with blue and yellow squares.

7. Conclusion and perspectives

We have proposed an algorithm of fuzzy segmentation, the first of its kind as far as we know. It performs a region growing on the image of gradient norms and builds a membership function equal to the length of the shortest path along the gradient image to the core of the region. It integrates the advantages of both approaches of image segmentation (region and edges) : it provides closed regions and performs a complete segmentation of the image, constrained by the large gradient values.

The interest of this method is not the segmentation itself, since it does not lead to an accurate partition of the image, but on the fuzzy, imprecise regions. They are useful for pattern recognition as well as for image indexing, because in these two domains it is more important to have region features, than an accurate segmentation with precise boundaries.

This algorithm can be applied to colour or grey level images. It can be used for the automatic segmentation of a whole database, since it requires no parameter, except an approximate number of expected regions, which will determine the degree of the segmentation. The user only indicates an interval for the expected number of regions

We have also developed an algorithm of inexact matching, adapted to image retrieval from partial query, made of several regions of an image. We have proposed solutions to reduce the combinatorial complexity of graph matching. This algorithm can be used with any segmented set of images, as soon as images are represented by attributed relational graphs. Inexact matching overcomes the problem of over or under segmentation, since one query region can match several target regions and vice-versa. The algorithm takes into account the similarity between regions, as well as their relative positions. We have showed on examples that our system is able to retrieve images containing a type of object or animal with a great variability of position (from the front, in profile, etc.) and of scale.

We showed results with a simple set of features (colour and texture distributions and mass centre position), and a simple distance. Spatial relations between regions are composed of the adjacency and the relative position of region mass centres. Other features and a more sophisticated characterisation of the relative positions of regions within the

image will certainly improve the results. For example a weighting related to the region size or to more accurate topological features can be added.

Applications may be of two types : An on-line search from a partial query, with a relevance feedback process aiming at interactively leading the search, and an off-line learning of categories from a set of examples of the object.

8. References

- [1] S. Agarwal, A. Awan, D. Roth, Learning to detect objects in images via a sparse, part-based representation, *IEEE Trans. On PAMI*, 26 (11), 1475-1490, 2004
- [2] J-F. Allen, Maintaining knowledge about temporal intervals, *Comm. of the ACM*, 11 (26), 832-843, 1983
- [3] M.C. De Andrade, G. Bertrand, A.A. Araujo, Segmentation of microscopic images by flooding simulation: A catchment basins merging algorithm. *Proc. of the IST/SPIE Symp. On Electronic Imaging, Science & Technology*, San Jose-CA, USA, 1997.
- [4] U. C. Benz, P. Hofmann, Gr. Willhauck, I. Lingenfelder and M. Heynen, Multi-resolution, object-oriented fuzzy analysis of remote sensing data for GIS-ready information, *IPRS Journal of Photogrammetry and Remote Sensing*, 53(3-4), 239-258
- [5] S. Berretti, A. Del Bimbo, E. Vicario, Efficient matching and indexing of graph models in content-based retrieval, *IEEE Trans. on PAMI*, 23 (10), 1089-1105, 2001
- [6] S. Berretti, A. Del Bimbo, E. Vicario, Weighted walkthroughs between extended entities for retrieval by spatial arrangement, *IEEE Trans. on Multimedia*, 5 (1), 52-70, 2003
- [7] J.C. Bezdek, *Pattern recognition with fuzzy objective function algorithms*, Plenum Press, New York, 1981.
- [8] J.C. Bezdek, J. Keller, Krisnapuram, S. K. Pal, *Fuzzy models and algorithms for pattern recognition and image processing*, ed. Kluwer, 2000.
- [9] N. Boujemaa, G. Stamon, J. Lemoine, Fuzzy iterative image segmentation with recursive merging. *SPIE 1818, Visual communication and image processing*, 1271-1281, 1992.
- [10] I. Bloch, Fuzzy relative position between objects in image processing : a morphological approach, *IEEE Int. Conf. on Image Processing*, Lausanne, 987-990, 1996
- [11] C. Carson, S. Belongie, H. Greenspan, J. Malik, Blobworld: image segmentation using Expectation-Maximisation and its application to image querying, *IEEE Trans. on PAMI*, 24 (8), 1026-1038, 2002
- [12] E Chang, B Li, G Wu, KS Goh, Statistical learning for effective visual information retrieval, *IEEE Int. Conf. on Image Processing*, Barcelona, 2003
- [13] S. K. Chang, Q. Y. Shi, C. W. Yan, Iconic indexing by 2-D strings, *IEEE Trans. on PAMI.*, 9 (3), 413-428, 1987
- [14] H.D. Cheng, Y-H. Chen, Fuzzy partition of two-dimensional histogram and its application to thresholding. *Pattern Recognition* 32, 825-843, 1999
- [15] J-P. Cocquerez, S. Philipp coordinateurs "Analyse d'images: filtrage et segmentation", ed *Masson*, sept. 1995
- [16] M. Cord, P-H. Gosselin, S Philipp-Foliguet, Stochastic exploration and active learning for image retrieval, *Image and Vision Computing*, 2006 (in press)
- [17] M Cord, S Philipp-Foliguet, P-H. Gosselin, J Fournier, Interactive exploration to image retrieval , *Journal of Applied Signal Processing*, vol 2005, issue 14, Special Issue on Advances in Intelligent Vision Systems: Methods and Applications—Part II, 2173-2186, 2005
- [18] L. P. Cordella, P. Foggia, C. Sansone, M. Vento, Subgraph Transformation for the inexact Matching of Attributed Relational Graphs, *Computing*, 12, 43-52, 1998.
- [19] A. Del Bimbo, M. Mugnaini, P. Pala, F. Turco, Visual Querying by colour perceptive regions, *Pattern Recognition*, 31 (9), 1241-1253, 1998
- [20] S. Di Zenzo, A note on the gradient of a multi-image, *CVGIP: Graphics and image processing* 33, 116-125, 1986

- [21] D. Dubois, M.C. Jaulent, A general approach to parameter evaluation in fuzzy digital pictures, *Pattern Recognition Letters* 6, 251-259, 1987
- [22] M. J. Egenhofer, R.D. Franzosa, Point-set topological spatial relations, *Int. J. for Geographical information systems*, 5(2), 161-174, 1991
- [23] M. Everingham, A. Zisserman, C. K. I. Williams, L. Van Gool, M. Allan, C. M. Bishop, O. Chapelle, N. Dalal, T. Deselaers, G. Dorkó, S. Duffner, J. Eichhorn, J. D. R. Farquhar, M. Fritz, C. Garcia, T. Griffiths, F. Jurie, T. Keysers, M. Koskela, J. Laaksonen, D. Larlus, B. Leibe, H. Meng, H. Ney, B. Schiele, C. Schmid, E. Seemann, J. Shawe-Taylor, A. Storkey, S. Szedmak, B. Triggs, I. Ulusoy, V. Viitaniemi, J. Zhang, The 2005 PASCAL Visual Object Classes Challenge Selected Proceedings of the first PASCAL Challenges Workshop, LNAI, Springer - 2006
- [24] J. Fauqueur and N. Boujemaa. Region-based image retrieval: Fast coarse segmentation and fine color description. *Journal of Visual Languages and Computing*, special issue on Visual Information Systems, 15(1), 69-95, 2004
- [25] P.H. Gosselin, M. Cord, RETIN AL : an active learning strategy for image category retrieval, *IEEE Int. Conf. on Image Processing*, Singapore, 2004
- [26] R. Krishnapuram, J. M. Keller, A probabilistic approach to clustering, *IEEE Trans. on Fuzzy Systems*, 1 (2), 98-110, 1993
- [27] R. Krishnapuram, J. M. Keller, Y. Ma, Quantitative analysis of properties and spatial relations of fuzzy image regions, *IEEE Trans. on Fuzzy Systems*, 1 (3), 222-233, 1993
- [28] I. Kompatsiris, E. Triantafillou, G. Strintzis, Region-based colour image indexing and retrieval, *IEEE Int. Conf. on Image Processing*, Thessaloniki, Greece, 658-661, 2001
- [29] P. Hong, T. S. Huang, Spatial pattern discovering by learning the isomorphic subgraph from multiple attributed relation graphs, *Electronic Notes in Theoretical Computer Science*, 46, Elsevier Science Publishers, Sébastien Fourey, Gabor T. Herman and T. Yung Kong ed, 2001
- [30] F. Jing, M. Li, H-J. Zhang, B. Zhang, An efficient and effective region-based image retrieval framework, *IEEE Trans. on PAMI*, 13 (5), 699-709, 2004
- [31] D. G. Lowe, Distinctive Image Features from Scale-Invariant Keypoints, *In. J. of Computer Vision*, 60(2), 91-110, 2004
- [32] W. Y. Ma, B. S. Manjunath, NeTra: a toolbox for navigating large image databases, *ACM Multimedia Systems*, 7 (3), 184-198, 1999
- [33] Y-F. Ma, H-J. Zhang, Contrast-based image attention analysis by using fuzzy growing, Proceedings of the 11th ACM International conference on Multimedia, Berkeley, CA, USA ,374 – 381, 2003
- [34] J. Malki, E. Zahzah, L. Mascarilla, Indexation et recherche d'image fondées sur les relations spatiales entre objets, *Traitement du signal*, 19 (4), 235-250, 2002
- [35] P. Matsakis, J.M. Keller, O. Sjahputera, J. Marjamaa, The use of force histograms for affine-invariant relative position description, *IEEE Trans on PAMI*, 26 (1), 1-18, 2004
- [36] S. Medasani, R. Krishnapuram, A fuzzy approach to content-based image retrieval, *IEEE Int.. Fuzzy systems Conf. Proc*, Seoul, 1251-1260, 1999
- [37] S Mika, G Ratsch, J Weston, B Scholkopf, KR Muller, Fisher discriminant analysis with kernels, *Advances in Neural networks for signal processing IX*, 41-48, 1999
- [38] A. Moghaddamzadeh, N. Bourbakis, A fuzzy region growing approach for segmentation of colour images. *Pattern Recognition*, 30 (6), 867-881, 1997.
- [39] N. R. Pak, S. K. Pal, A review on image segmentation techniques. *Pattern Recognition* 26 (9), 1277-1294, 1993.
- [40] E.J. Pauwels, G. Frederix, Finding Salient Regions in Images: Non-Parametric Clustering for Image Segmentation and Grouping, *Journal of Computer Vision and Image Understanding*, 75, 73-85, 1999
- [41] A. Perchant, I. Bloch, Fuzzy morphisms between graphs, *Fuzzy Sets and Systems*, 128 (2), 2002
- [42] S. Philipp-Foliguet, M. B. Vieira, A. de A. Araújo, Segmentation into fuzzy regions using topographic distance, 14th SIBGRAPI, 282-288, Florianopolis, Brazil, 2001
- [43] S. Philipp-Foliguet, G. Logerot, P. Constant, PH. Gosselin, C. Lahanier, Multimedia indexing and fast retrieval based on a vote system, *ICME 06*, Toronto, 2006
- [44] B. G. Prasad, K. K. Biswas, S. K. Gupta, Region-based image retrieval using integrated color, shape and location index, *CVIU*, Special issue on colour for image indexing and retrieval, vol 94, 193-233, 2004

- [45] A. Robles-Kelly, E. R. Hancock, Graph matching using adjacency matrix Markov chains, Proc. 3rd Int. Workshop EMMCVPR, Sophia-Antipolis, France, 2001
- [46] A. Rosenfeld, The fuzzy geometry of image subsets. *Pattern Recognition Letters* 2, 311-317, 1984.
- [47] A. Rosenfeld, A., Hummel, R.A., and Zucker, S.W., Scene Labeling by Relaxation Operations, *IEEE Trans on SMC*, 6(6), 420-433, 1976
- [48] E.H. Ruspini, A new approach to clustering. *Information and Control*, 15 (1), 22-32, 1969.
- [49] A.W.M Smeulders, M. Worring, S. Santini, A. Gupta, R. Jain, Content-Based Image Retrieval at the End of the Early Years, *IEEE Trans on PAMI*, 22 (12), 1349-1380, 2000
- [50] J. R. Smith, S-F. Chang, VisualSEEk: a fully automated content-based image query system, *ACM Multimedia Conf.*, Boston, USA, 87-98, 1996
- [51] A. Steudel, M. Glesner, Fuzzy segmented image coding using orthonormal bases and derivative chain coding. *Pattern Recognition* 32, 1827-1841, 1999.
- [52] N. Vasconcelos, Bayesian Models for Visual Information Retrieval, PhD thesis, Massachusetts Institute of Technology, 2000
- [53] L. Vincent, P. Soille, Watersheds in digital spaces: an efficient algorithm based on immersion simulation, *IEEE Trans on PAMI*, 13 (6), 563-598, 1991.
- [54] F. Wang. Fuzzy supervised classification of remote sensing images, *IEEE Trans. on Geoscience and Remote Sensing*, 28(2):194--201, 1990.
- [55] J. Z. Wang, J. Li, G. Wiederhold, SIMPLiCity: semantics-sensitive integrated matching for picture libraries, *IEEE Trans. on PAMI*, 23 (9), 1-17, 2001