

From Content-Based Image Retrieval by Shape to Image Annotation

Irina MOCANU

University Politehnica of Bucharest, Computer Science Department
Splaiul Independentei str. no. 313, RO-0600429 Bucuresti, Romania
irina.mocanu@cs.pub.ro

Abstract—In many areas such as commerce, medical investigations, and others, large collections of digital images are being created. Search operations inside these collections of images are usually based on low-level features of objects contained in an image: color, shape, texture. Although such techniques of content-based image retrieval are useful, they are strongly limited by their inability to consider the meaning of images. Moreover, specifying a query in terms of low level features may not be very simple. Image annotation, in which images are associated with keywords describing their semantics, is a more effective way of image retrieval and queries can be naturally specified by the user.

The paper presents a combined set of methods for image retrieval, in which both low level features and semantic properties are taken into account when retrieving images. First, it describes some methods for image representation and retrieval based on shape, and proposes a new such method, which overcomes some of the existing limitations. Then, it describes a new method for image semantic annotation based on a genetic algorithm, which is further improved from two points of view: the obtained solution value – using an anticipatory genetic algorithm, and the execution time – using a parallel genetic algorithm.

Index Terms—image annotation, image representations, image retrieval by content, genetic algorithm, shape retrieval.

I. INTRODUCTION

Large image databases are used in many types of multimedia applications such as science, engineering, entertainment, and business. Due to the huge number of images in such databases, search and retrieval of images become a real challenge. One solution to efficient image retrieval is content-based retrieval, in which images are located based on different techniques for comparing the image content.

The earliest use of the term content-based image retrieval (CBIR) in the literature seems to have been by [1], to describe his experiments in automatic retrieval of images from a database. The retrieval was made based on the image content, which was represented by the low level features: colour, shape, or texture. CBIR differs from classical information retrieval in that image databases are essentially unstructured, since digitized images consist purely of arrays of pixel intensities, with no inherent meaning. One of the key issues with any kind of image processing is the need to extract useful information from the raw data (such as recognizing the presence of particular shapes or textures)

The work has been co-funded by the Sectoral Operational Programme Human Resources Development 2007-2013 of the Romanian Ministry of Labour, Family and Social Protection through the Financial Agreement POSDRU/89/1.5/S/62557 and Grant IDEI, ID_1315.

Digital Object Identifier 10.4316/AECE.2010.04008

before any kind of reasoning about the image's contents is possible.

There are two general methods for comparing images: intensity-based (color and texture) and geometry-based (shape). Based on a survey about cognitive aspects of image retrieval, [2] show that users are more interested in retrieving images by shape than by color and texture.

The effectiveness of all current CBIR systems is inherently limited by the fact that they can operate only at the primitive feature level. For example, a query for “sky” will return scenes of both daytime (where sky is mostly blue) and sunsets (where sky tends to be orange) with equal ease. It is usually quite difficult to design a set of visual features that captures all the relevant dimensions of image variability (e.g., that the sky can be either blue or orange). Thus combining primitive image features with text keywords can overcome some of these problems, therefore image annotation (a mapping between the regions of an images and a set of words) becomes an important operation towards describing the semantics of an image..

The aim of this paper is to propose a combined set of methods for image retrieval, in which both low level features and semantic properties are taken into account when retrieving images. The first part of the paper (Section II.) describes and compares four boundary based methods for shape representation and retrieval: Fourier descriptors-based method, turning angle method, centroid-radii method, and distance histogram method. A new method, called “centroid radii and turning angle” is proposed. The five methods are compared by precision and recall, processing time, and comparison time. The second part of the paper (Section III.) presents a method for semantic annotation of objects in images by first performing vocabulary classification (using k-means clustering) and then applying a standard genetic algorithm for associating keywords. Moreover, two variants of the standard genetic algorithm are proposed, aiming to improve the results obtained by image annotation.

II. IMAGE RETRIEVAL BASED ON SHAPE

Shape is an important low-level image feature. For retrieval based on shapes, images must be first segmented into individual objects. After segmentation, the basic issue of shape-based image retrieval is shape representation and similarity measurement between shape representations.

Image retrieval systems are based on best-match techniques instead of exact match techniques. During the retrieval process, the user selects an example image that he

or she is interested in. The image is then presented to the retrieval system. The system retrieves images containing similar shapes and presents them to the user in the order of decreasing similarity – the similarity measure between shapes representations should conform to human perception.

A good shape representation and similarity measurement for recognition and retrieval purposes should have the following two important properties [3]: (i) each shape should have a unique representation, invariant to translation, rotation, and scale; (ii) similar shapes should have similar representations so that retrieval can be based on distances among shape representations.

A. Fourier Descriptors-Based-Method

Reference [4] proposed the technique of using the Fourier descriptors (FD) as the representation of shapes. Fourier descriptors as in [5] are obtained by applying Fourier transform to shape boundary (represented by a shape signature). In [6], the authors show that shape signature using centroid distance has good results in shape based retrieval.

The centroid distance function (r_i) is expressed by the Euclidean distance of the boundary points to the centroid of the shape. All the shapes are normalized to the same number of boundary points N . The discrete Fourier transform of is then given by:

$$u_n = N^{-1} \sum_{i=0}^{N-1} r_i \exp(-j2\pi niN^{-1}), n = 0, 1, \dots, N-1. \quad (1)$$

The coefficients u_n , $n = 0, \dots, N-1$, are called Fourier descriptors (FD) of the shape: FD_n , $n = 0, 1, \dots, N-1$.

Shape radii and thus their transformations are translation invariant. $|FD_n|$, is invariant to rotation and $|FD_0|$ reflects the energy of the shape radii, thus $|FD_n|/|FD_0|$ will be scale invariant. The vector $f = [|FD_1|/|FD_0|, |FD_2|/|FD_0|, \dots, |FD_{N-1}|/|FD_0|]$ is used as the Fourier descriptors.

The similarity measure of the query shape (A – described by the Fourier descriptors $FDA: (FD^A_1, \dots, FD^A_{N-1})$) and a target shape from the database (B described by the Fourier descriptors $FDB: (FD^B_1, \dots, FD^B_{N-1})$) is the Euclidean distance between the query and the target shape feature vectors as in:

$$d(A, B) = \sqrt{\sum_{i=1}^{N-1} |FD^A_i - FD^B_i|^2}. \quad (2)$$

B. Turning Angle Method

A boundary of a polygon A is described by a list of vertices (x_i, y_i) . In the Turning Angle (TA) method, presented in [7], the boundary is described by the turning function $\theta_A(s)$, which measures the angle of the counter clockwise tangent as a function of the arc length s . The angle is measured from a given reference start point O on the polygon's orientation, i.e. the x axis.

This type of representation of shape is invariant to translation. To make it invariant to scale, the polygon will be normalized such that its perimeter length will be 1. The representation is not invariant to rotation. Consider two polygons A and B with their turning angle $\theta_A(s)$ and $\theta_B(s)$, and rotate one of them, for instance B , by a certain angle α ;

then the new $\theta^*_B(s)$ is only a simple vertical shift in y axis of $\theta_B(s)$. To measure the similarity between the two objects without being affected by rotations, all possible vertical translations for the $\theta_A(s)$ with α , (α ranging between $[0, 2\pi]$), should be considered.

Another factor to consider is the arbitrary nature of the starting point along the boundary of the object; the vectors $\theta_A(s)$ and $\theta_B(s)$ may be the same unless there is an arbitrary horizontal translation of the samples (a circular shifting of the turning angle vector elements).

To compare two shapes by their turning angles vectors, the Euclidean distance is used. For two shapes A and B whose turning functions are $\theta_A(s)$ and $\theta_B(s)$, the distance function has to take into account all the possible vertical and horizontal translations of the samples of A and B . The distance between two shapes A and B will be compared using:

$$d(A, B) = \min_{r, \theta} \sqrt{\sum_i (\theta_A(i) - \theta_B(i))^2}. \quad (3)$$

where $\min_{r, \theta}$ represents the minimum of all possible horizontal and vertical translations of θ_B .

C. Centroid-Radii Method

In [8] a method for shape representation based on centroid-radii model is proposed (the CR method). The shape is represented by the lengths of the shape's radii from centroid to boundary. The interval between radii, θ , is fixed. Therefore, the number of intervals can be calculated by $k = [360/\theta]$. The shape can then be represented by the vector: $(L_0, L_0, \dots, L_{(k-1)\theta})$ (where $L_{i\theta}$, $0 \leq i \leq k-1$ is the $(i+1)^{th}$ radius from the centroid to the boundary of shape). This representation is invariant to translation. To make this representation invariant to scale, all radii lengths can be normalized by dividing the longest radius among them and obtain: $(l_0, l_0, \dots, l_{(k-1)\theta})$ where $l_{i\theta}$ are the normalized values. In most cases the representation itself is not robust to rotation. When a different starting point is chosen, a new radius vector is obtained.

The distance between the two shapes A and B will be computed using the Manhattan distance between the two feature vectors as in:

$$d(A, B) = \sum_{i=0}^{k-1} \min_{r, \theta} |l^A_{i\theta} - l^B_{i\theta}|. \quad (4)$$

D. Distance Histogram Method

The Distance Histogram (DH) method, proposed in [9], is similar to the centroid-radii method [8]. This method computes a histogram based on the length of the shape's radii measured from the shape's centroid to its boundary. A number of sample points for each edge evenly spread on it are chosen. This number is proportional with the length of each edge referred to the perimeter of the shape.

The radii are computed using the Euclidean distance between the sample points and the centroid of the shape. The distance histogram will be computed based on the set of radii $\{d_1, d_2, \dots, d_n\}$. This approach is invariant to translation. The sample points for each edge are evenly

spread on it. Thus the method is invariant to rotation. For two shapes A and B, where B is a scaled version of A, the radii set of B will be different from the radii set of A. So the DH is not invariant under scale. The scaling normalization consists of dividing the value of the lengths of all radii by the value of the maximum length of the radii length.

The range of the length of all radii $[0, R_{\max}]$ is separated into R ranges. Then, the distance histogram is represented as: $D: (d_0, d_1, d_2, \dots, d_{R-1})$, where $d_i, i = 0 \dots (R-1)$ is the number of distances belonging to i^{th} distance range.

The similarity between two shapes A and B with the distance histograms $D^A: (d^A_0, d^A_1, d^A_2, \dots, d^A_{R-1})$ and $D^B: (d^B_0, d^B_1, d^B_2, \dots, d^B_{R-1})$ is given by the Euclidean distance:

$$d(D^A, D^B) = \sqrt{\sum_{i=0}^{R-1} (d^A_i - d^B_i)^2}. \quad (5)$$

E. Centroid Radii and Turning Angle Method

Let us consider the two shapes from Fig. 1. The distance between the two shapes (computed with (5)) is nearly 0 (the lengths of the radii associated with each shape are very closely), which means that the two shapes are similar based on the DH method, although they look different. This happens because the distance histogram method discards spatial information to obtain rotation invariant.

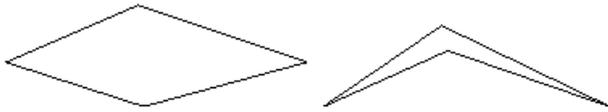


Fig. 1. Two similar shapes based on DH method

To consider spatial representation, a new method for shape representation and retrieval – the Centroid Radii and Turning Angle (CRTA) method was proposed in [10]. A shape is represented by the directions of the edges together with the radii list: (a) the direction of an edge is represented by the turning angle of the edge (as in TA method); (b) each edge has associated a list of corresponding radii (the corresponding radii are obtained as in the DH method).

This representation is invariant to translation, but it is not invariant to scale. To make it scale invariant, all the distances between the centroid of the shape and the chosen sample points must be normalized: dividing the value of all distances by the value of the maximum distance. The representation is not invariant to rotation. For a rotated shape, all the edges' directions will increase or decrease with the same rotation angle. Thus all possible vertical translations for the edges' directions with α , (α ranging between $[0, 2\pi]$), should be considered.

To compare two shapes using this method, the lists of directions must be compared, and the radii associated with them must be compared, too. In order to compare the radii associated with each direction, the standard deviation of the radii are calculated. Therefore a shape will be represented by $\{(d_1, \text{std}_1), (d_2, \text{std}_2), \dots, (d_n, \text{std}_n)\}$, where d_1, d_2, \dots, d_n represent the edges' directions of the shape and $\text{std}_1, \text{std}_2, \dots, \text{std}_n$ represent the standard deviation of each direction.

In this case one shape can have more edge directions than the other. But we cannot say that the shapes are different because the shapes can be affected by noise. Thus a supplementary stage is necessary, in which the boundary of

the shape is smoothed using an iterative algorithm (with linear complexity on the number of boundary points [11]).

Thus the similarity between two shapes A and B represented by $\{(d^A_1, \text{std}^A_1), (d^A_2, \text{std}^A_2), \dots, (d^A_n, \text{std}^A_n)\}$ and $\{(d^B_1, \text{std}^B_1), (d^B_2, \text{std}^B_2), \dots, (d^B_m, \text{std}^B_m)\}$, respectively, is given by:

$$d(A, B) = \text{direction_dist} + \text{radii_dist}. \quad (6)$$

where direction_dist is the distance between the edges' directions, and radii_dist is the distance between the radii associated with each direction.

$$\text{direction_dist} = \min_k \sqrt{\sum_{i,j} (d^A(i) * L_r - d^B(j) * L_r)^2}. \quad (7)$$

and

$$\text{radii_dist} = \sqrt{\sum_{i,j} (\text{std}^A(i) * L_r - \text{std}^B(j) * L_r)^2}. \quad (8)$$

where \min_k represents the minimum of all circular shifted of d^A , L_r is the length of the common edge for i direction, respectively j direction of the two shapes and $\text{std}^A(i)$, $\text{std}^B(j)$ is the standard deviation of the radii associated with the i^{th} , respectively j^{th} direction of the two shapes (they correspond to the circular shifted position of the edges' direction for direction_dist is minimum).

F. Experimental Results

In this section, a performance comparison of the five discussed methods is presented. To test the retrieval performance of these methods, a retrieval framework has been implemented on two databases with synthetic shapes:

- the database A with simple polygonal shapes described by boundary points (maximum 20 boundary points): approximately 400 polygons (without noise);
- the database B with synthetic shapes of fishes used by SQUID system [12]: approximately 2000 synthetic shapes: there are 500 initial shapes (the shapes may be affected by noise) Each shape is described by the boundary points, too. On the average, there are 700 boundary points.

The performance has been evaluated using:

- precision and recall as described in [3]; Precision is defined as the ratio of the number of similar shapes retrieved to the total number of shapes retrieved. Recall is defined as the ratio of the number of similar shapes retrieved to the total number of similar shapes in the whole database. Precision indicates accuracy of the retrieval and recall indicates the robustness of the retrieval performance.
- processing time (time for computing all of the information used to compare two shapes);
- comparison time (time for comparing two shapes).

The query shape is selected from the database.

Considering the average precision and recall (Fig. 2.) obtained for the two test databases, the centroid radii and turning angle method outperforms the rest of the analyzed methods. The low retrieval performance of turning angle

method indicates it as an inaccurate shape representation. There are few situations in which distance histograms method outperforms precision and recall of the centroid radii and turning angle method (these situations appear for shapes with a lot of small edges).

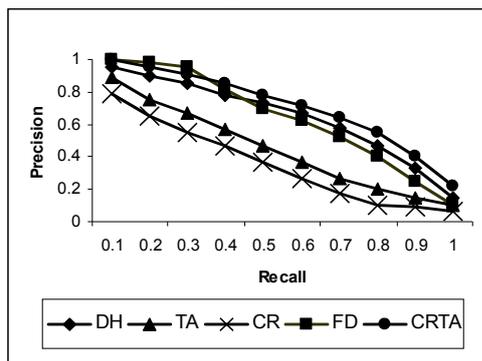


Fig. 2. Average precision and recall

The average processing time and average comparison time for the described methods are given in Fig. 3a. and Fig. 3b., respectively. The CR method has the lowest processing time, but the CRTA method has the longest processing time. The CR method needs to calculate only the centroid radii for representing a shape while the CRTA method needs to compute more data – (edges' directions and centroid radii associated with every edge).

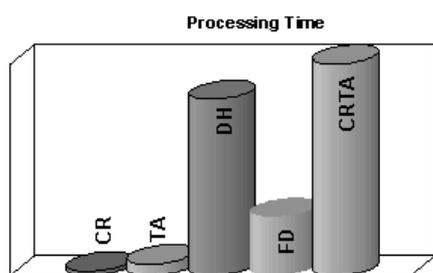


Fig. 3a. Average processing time (CR \sim 0.02 CRTA, TA \sim 0.06 CRTA, FD \sim 0.3 CRTA, DH \sim 0.9 CRTA)

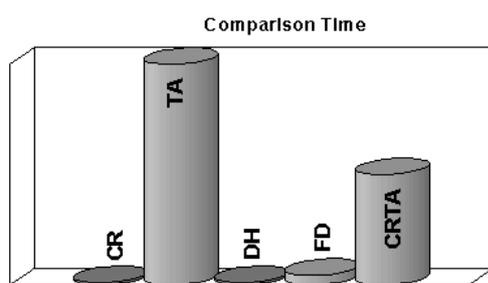


Fig. 3b. Average comparison time (CR \sim 0.01 TA, DH \sim 0.01 TA, FD \sim 0.05 TA, CRTA \sim 0.5 TA)

The CR method and DH method have the lowest comparison time, but the TA method has the longest comparison time. The CR method and Fourier descriptor-based method represent shapes by feature vectors with similar number of components. The TA method represents shapes using the largest number of components for feature vectors. On the average, the comparison time for CRTA method is two times lower than the comparison time for TA method. But the comparison time for CRTA method is very high compared to the other presented methods: FD, CR and DH methods.

Based on the results from Fig. 2. (precision – recall graph) the performance of the CRTA method is higher than the performances of the other four methods. Using a preprocessing step for smoothing the edges of the shape, as described in [13], the comparison time for CRTA method is reduced by a factor of 4, showing that the CRTA method achieves good results for image retrieval.

III. IMAGE ANNOTATION

The requirements of image users can vary considerably; these requirements can be used to classify image queries into three levels of abstraction of increasing complexity as in [14], [15]:

- Level 1 comprises retrieval by primitive features such as color, texture, shape or the spatial location of image elements. For example: “find pictures with long thin dark objects in the top left-hand corner”, “find images containing yellow stars arranged in a ring” – or most commonly “find me more pictures that look like this”. This level of retrieval uses features (such as a given shade of yellow) that are both objective and directly derivable from the images themselves, without the need to refer to any external knowledge base.
- Level 2 comprises retrieval based on derived features (or logical features), involving some degree of logical inference about the identity of the objects depicted in the image. It can be: (i) retrieve objects of a given type (e.g., “find pictures of a double-decker bus”); or (ii) retrieve individual objects or persons (“find a picture of the Eiffel tower”).
- Level 3 comprises retrieval by abstract attributes, involving a significant amount of high-level reasoning about the meaning and purpose of the depicted objects or scenes. It can be: (i) retrieve of named events or types of activity (e.g., “find pictures of Scottish folk dancing”); or (ii) retrieve pictures with emotional or religious significance (“find a picture depicting suffering”).

Many authors, for example [16], refer to levels 2 and 3 of image queries as semantic image retrieval, and identify a gap between levels 1 and 2, called the semantic gap.

Image manual annotation is a difficult process. A large number of techniques have been proposed to resolve image annotation automatically. Many of these techniques treat the problem of image annotation as a translation from image instances (object from images) to keywords, and are based on unsupervised or supervised machine learning techniques, as described for example in [17], [18] [19].

A. Image Annotation Using a Traditional Genetic Algorithm

Based on an idea of [20], we present a new image annotation technique based on a genetic algorithm (GA). The proposed algorithm has the structure of a classical GA and is called Image Annotation Genetic Algorithm (IA_GA).

The vocabulary used for image annotation consists of a set of words and each word is associated with a representative image - each representative image for a word from vocabulary contains one meaningful region/segment. All these images will be named vocabulary images. In our

approach, image annotation is based on the color and shape similarities. The color similarities are obtained using the histogram color method. The used color space is HSV space based on the results from [21] – a model based on the artist concepts: Tint, Shade and Tone. The distance between two objects from the point of view of the color low-level feature is obtained by the histogram intersection, proposed in [22].

As opposed to the method described in [20], our proposed approach classifies the vocabulary images starting from their shapes and using the k-means clustering algorithm [23].

The k-means clustering algorithm classifies a given data set into groups (named clusters) of similar data. The numbers of clusters are fixed a priori. Let us consider a given data set $D = \{x_1, \dots, x_N\}$ which we want to classify in k clusters. A set of centroids $C = \{C_1, \dots, C_k\}$ will be assigned to each cluster (the initial centroids may be randomly chosen from the data set). Each item x_i from the data set D will be associated to a cluster j (the cluster with the minimum distance between x_i and C_p , $p \in \{1, \dots, k\}$). When no point is pending, the first step is completed and an early grouping is done. At this point we need to re-calculate k new centroids of the clusters resulting from the previous step. After we have these k new centroids, a new binding has to be done between the same data set points and the nearest new centroid. After that the k centroids change their location step by step until the centroids do not move any more.

Each word from vocabulary will be represented by the shape extracted from the described image – the shape is described using the CRTA method, described in Section II.E. The k-means algorithm will be applied for image classification based on shape. The distance used in the k-means algorithm is the distance from (7) – by which are computed the similarity of two objects in CRTA method. Thus, at the end of the classification phase, each cluster will contain words which have similar shape.

Thus a region R_i will be compared only with a part of the vocabulary images, namely only with the images from the cluster to which R_i is closest (the cluster which has the minimum distance between R_i and its centroid).

The genetic algorithm will associate a set of representative words to each image. In fact each word will describe a region from the image. A chromosome in the genetic algorithm must reflect the similarity between the image and the current description of it. This similarity will be computed by the fitness function using the low level features. But the fitness function cannot be only a distance between low level features. It must reflect how good or bad is the current image description. Thus the fitness function must combine this distance with a similarity degree for the worst description from the used vocabulary. The obtained value will be mapped to the range from the worst similarity degree to the better similarity degree.

Before applying the genetic algorithm for image annotation, a preprocessing phase is needed. This phase comprises the following steps:

- each image is segmented into component regions R_1, R_2, \dots, R_N . In fact each region from an image is an object (each object is identified by its boundary).
- a similarity degree S_{ij} is assigned to each region R_i . S_{ij} is based on the similarity of the low level features (color and shape) between region R_i and object O_j ,

where R_i belongs to cluster p of the vocabulary and O_j is every object of cluster p . The color feature and the shape feature are considered in equal ratios – and normalized in interval $[0, 1]$. Each region R_i will have an associated set of similarity degrees: Set_i . We note S_{i_min} and S_{i_max} minimum and maximum similarity degree of region R_i .

The IA_GA algorithm for annotating an image containing N objects uses the following representation and genetic features:

- a chromosome in IA_GA is represented by N values $\{V_1, V_2, \dots, V_N\}$, where V_i represents the word W_{Ni} associated with region R_i as depicted in Fig. 4. When the algorithm terminates, W_{Ni} will annotate the region R_i . The initial population is generated randomly, based on the vocabulary used for annotation.

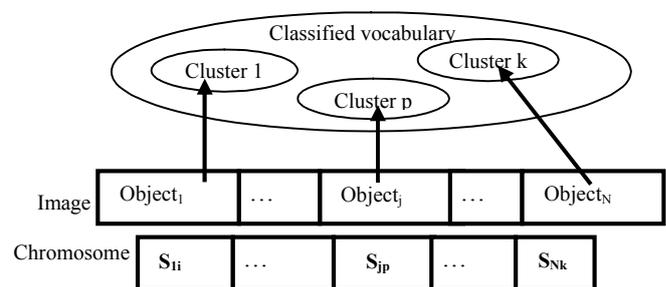


Fig.4. - The structure of a chromosome

- the fitness function for IA_GA must express the similarity degree between an object and the image associated with its current associated keyword from the vocabulary. The fitness function is based on the similarity between each region from an image and the current assigned word (S_{ij}).

$$fitness = \frac{\sum_{i=1}^N S_{ij} - \sum_{i=1}^N S_{i_min}}{\sum_{i=1}^N S_{i_max} - \sum_{i=1}^N S_{i_min}} \quad (9)$$

- selection is performed using a tournament selection operator;
- crossover is uniform crossover;
- mutation is achieved by an operation similar to bit flip mutation: the corresponding value of a gene is modified with a value associated to an image from the same cluster.

The IA_GA is evaluated with a small set of images from the flower domain (approximately 50 images) and a vocabulary with concepts: rose, lily, sunflower and lotus. For this vocabulary and test set, the IA_GA algorithm time needs to compute the set of similarity degree for each region from an image is reduced by a factor of 0.3 as compared to the method presented in [20].

The following genetic parameter values are considered to test the performances of the IA_GA: (i) 20 individuals in each population; (ii) 100 generations; (iii) 40 test runs for each image. The results show that the IA_GA algorithm achieves satisfactory associations of image objects to

significant keywords.

B. Image Annotation Using an Anticipatory Genetic Algorithm

A genetic algorithm with anticipative behavior, called Anticipatory Genetic Algorithm (AGA), was introduced in [24]. AGA's central idea is that each new generation should be closer to the solution than the previous one. In order to ensure this important property, candidate chromosomes that are worse, from the fitness point of view, than the worst candidates in the previous generation are rejected. Consequently, each population is characterized by its Evolution Threshold (ET), which is represented by the worst fitness value (WFFV) computed for its members. According to the particular problem solved, WFFV is either the lowest or the highest fitness value in the considered population.

This evolution property is achieved by filtering the new generated chromosomes, which are either included in population P_{i+1} or rejected. In order to compensate the rejection effect, each pair of selected parent chromosomes generates 4 or more offsprings: the first two by crossover and the next two as mutated clones of the first ones.

Combining AGA with the IA_GA algorithm leads to a new modified genetic algorithm for image annotation, that we call IA_AGA. As in AGA, for each new generation four children (chromosomes) are created: two obtained by crossover, and two others obtained as mutated clones of the first ones. The evolution threshold for a new generation $i+1$ (ET_{i+1}) is represented by the maximum fitness value of the individuals from generation i . A newly generated chromosome will be considered appropriate for the next generation and will be included in the P_{i+1} population only if its fitness value is less than ET_i . The AGA algorithm is presented in Fig. 5.

```

{ generate initial population ( $P_0$ );
  evaluate the fitness  $f(x)$  for each chromosome;
  compute current evolution threshold ( $ET_0$ )
   $i = 0$ 
  while  $i < No\_generations$  do
  { initialize new population ( $P_{i+1}$ ) and
    new evolution threshold ( $ET_{i+1}$ )
    repeat
    { select parent chromosomes
      apply crossover operator to generate the
        primary 2 offsprings
      apply mutation operator to clones of
        the primary offsprings to generate 2
        or more mutants
      for each offspring  $y$  repeat
      { evaluate fitness  $f(y)$ 
        if Acceptable( $y, ET_i$ ) then
        { include  $y$  in  $P_{i+1}$ 
          update  $ET_{i+1}$ 
          if Full( $P_{i+1}$ ) then break
        }
        else Reject( $y$ )
      }
    }
  }until Full( $P_{i+1}$ )
  increase  $i$  ( $P_{i+1}$  becomes  $P_i$ )
}

```

Fig. 5. Anticipatory genetic algorithm for image annotation.

The IA_AGA is tested using the same test conditions as those described in Section III.A. For each image, the IA_GA and IA_AGA are run and the obtained keywords are compared with the component object. The comparison is

realized based on the fitness of the obtained solution, which represents the distance between the low level features of the object to be annotated and the annotation. On the average, the results obtained with the IA_AGA algorithm are 5% better than in the case of IA_GA, making the IA_AGA annotations more accurate. Although better from the point of view of the obtained results, the IA_AGA puts more effort to obtain the solution: it needs 2 times more generated chromosomes until reaching a solution than IA_GA does, leading thus to an increased execution time of about 1.8.

C. Parallel Genetic Algorithm

In order to keep both the good annotation results generated by IA_AGA and the computing performances of IA_GA, we developed an parallel implementation of the genetic algorithm used in IA_AGA. The way in which a genetic algorithm can be parallelized depends on the following elements [25]:

- how fitness is evaluated and mutation is applied;
- if single or multiple subpopulations are used;
- in case multiple populations are used, which is the scheme for individuals exchanged;
- how selection is applied (globally or locally).

Depending on how we choose these parameters of the parallel GA, more than ten different methods of parallelizing genetic algorithms can be obtained [25]. One of these classes of parallel genetic algorithms is called static subpopulation with migration. The important characteristics of this class of parallel GA are the use of multiple subpopulations and the presence of a migration operator. This parallelization method requires the division of the population into some number of subpopulations. Subpopulations are separated from one another (geographic isolation), and individuals compete only within a subpopulation. An additional operator, called migration, is introduced. From time to time, the migration operator moves (copies) some individuals from one subpopulation to another. The migration of individuals from one subpopulation to another is controlled by several parameters:

- the topology that defines the connections between the subpopulations; for example: complete topology, ring topology, two-dimensional/three-dimensional mesh topology, etc.
- a migration scheme that controls which individuals from the source subpopulation (best, worst, random) migrate to another subpopulation, and which individuals are replaced (worst, random, etc.).
- a migration interval that determines the frequency of migrations.

The IA_AGA algorithm is modified to account for the parallel implementation of the GA and we call the resulting algorithm IA_PAGA. The initial population is divided in T subpopulations: $P_{01}, P_{02}, \dots, P_{0T}$. Each population runs and evolves on a thread, using a total of T threads. If N is the number of individuals of the initial population, then the subpopulation on the threads 1 ... ($T-1$) will contain N/T individuals and the last thread will contain a subpopulation of $(N-[N/T]*(T-1))$ individuals. Each subpopulation will evolve independently. The threads will be synchronized from time to time. At the synchronization time the

subpopulations will exchange some individuals.

For fine tuning the algorithm, the specific parameters of the parallel GA must be chosen. For the anticipatory genetic algorithm for image annotation presented in Section III.B, the evolution threshold (the maximum fitness value of the individuals from a subpopulation in a generation) is needed in every generation. The next population is constructed keeping the best chromosomes from the current population. Thus, all the subpopulations must communicate and exchange their evolution threshold and their best chromosomes - the number of chromosomes exchanged will be equal to the number of chromosomes on which elitism selection is applied. The chosen topology must be a complete one. The subpopulations will exchange.

The IA_PAGA is presented in Fig. 6.

```

IA_PAGA ()
{ generate initial population (P0);
  evaluate the fitness f(x) for each chromosome;
  compute the initial evolution threshold (ET0)
  split P0 into T subpopulations: P0_0, P0_1, ..., P0_T
  i = 0
  repeat
  {
    parallel
    {
      for(j = 1; j <= T; j++) Run_Thread(j);
      Generate_sincronization();
      Migrate();
    }
    end parallel
  }until i > No_generations
}

```

```

Run_Thread(j)
{ i = 0;
  repeat
  { initialize new population (P(i+1)_j);
    initialize new evolution threshold (ET(i+1)_j)
    repeat
    { select parent chromosomes
      apply crossover operator to generate the
        primary 2 offsprings
      apply mutation operator to clones of the
        primary offsprings to generate 2 or
        more mutants
      for each offspring y
      {
        if Acceptable(y, ETi_j) then
        { include y in P(i+1)_j
          update ET(i+1)_j
          if Full(P(i+1)_j) then break
        }
        else Reject(y)
      }
    }until Full(P(i+1)_j)
    increase i (P(i+1)_j becomes Pi_j)
  }
}

```

```

Generate_sincronization()
{
  generate a synchronization step for the T
  threads after each migration interval
}

```

Fig.6. The parallel anticipatory genetic algorithm.

The specific parameters of the IA_PAGA are:

- the topology for connections between the subpopulations: complete topology;
- the migration scheme: the individual with the maximum fitness value (the evolution threshold) and

the best individuals (with minimum fitness value) from a subpopulation – are exchanging;

- the number of subpopulations and the migration interval will be chosen based on the experimental tests.

Each subpopulation will evolve independently running on a thread. The threads will be synchronized from time to time. At the synchronization time, the subpopulations will exchange their evolution threshold and their best chromosomes (e chromosomes). A vector V is used for exchanging these values. When the first thread enters into the synchronizing step it places in vector V its evolution threshold on the first position, and on the next e positions – the best chromosomes from its population. The following threads, when entering the synchronization step, compare their evolution threshold with the value from the first position of vector V. Also, their best chromosomes are compared with the ones already present in the vector V (the values of vector V from position 1 to e). If some of their values are better than the values founded in vector V, they will replace the values in the vector V. When the last thread finishes this job, all threads meet at a synchronization point and after releasing, they will copy the evolution threshold and the chromosomes from the vector into their populations. Fig. 7 outlines the algorithm used for exchanging best chromosomes together with the evolution threshold.

```

Migrate()
{ if(Thread_entered == 1)
  {
    set ET with the evolution threshold of
    the current thread put the best e
    chromosomes in vector V
  }
  else
  {
    if(ET < V[0]) set V[0] with ET
    i = 1;
    while(i <= e)
    {
      if(fitness of chromosome[i] < fitness V[i])
        set V[i] with chromosome[i]
      i++;
    }
    barrier();
    for(i = 0; i < e; i++) chromosome[i] = V[i];
    set the evolution threshold of the current
    thread with ET
  }
}

```

Fig. 7. - The migration scheme for the IA_PAGA.

D. Performance Evaluation IA_PAGA

The IA_PAGA is tested in the same conditions as IA_AGA: the same vocabulary is used for annotation, and the same parameters are selected for the genetic algorithm, namely the number of individuals in the population, the number of generations, and the number of tested images. Each image is annotated with IA_AGA and with IA_PAGA. In the case of IA_PAGA, the values for the following two parameters must be chosen: the number of subpopulations and the migration interval. These two parameters will be varied such that the accuracy of the annotation remains unchanged. At the same time, these two parameters are chosen in such a way that the execution time is reduced so as to be comparable to the one of IA_AGA. Benchmarking

the same images as in IA_AGA, the best speed gain was obtained for 2 subpopulations and migration interval used after every 25 generations. In this case, the execution time was reduced by 4.6%.

IV. CONCLUSION

The main contribution of this paper was to show how the combination of low level features of images (color and shape) with their semantic content can be used for image retrieval. Towards this aim, the paper proposes a set of algorithms that try to bridge the semantic gap between the level 1 and levels 2, 3 in image queries. The first part of the paper presented some methods for image retrieval based on a low level feature, the shape: DF, TA, CR, DH, CRTA methods. The five methods were compared based on precision and recall, the processing time, and the comparison time. As shown by the precision and recall graph, the CRTA method was evaluated to have the best retrieval performance, while the TA method to have the worst performance. Based on the performances obtained by the CRTA method, this method was used for obtaining the semantic aspects of an image. Two low-level features, color and shape, are used to annotate an image. The second part of the paper described a standard genetic algorithm for image annotation based on image representation by color and shape. The genetic algorithm uses the CRTA method for similarity determination and has been modified with an anticipatory behavior, to increase the performance of image annotation. Using the anticipatory genetic algorithm, the obtained solution value is better but the execution time is increased. Thus, a parallel genetic algorithm was proposed, leading to an improvement of the execution time of 4.6%.

As future work, the proposed genetic algorithm will be integrated in an enhanced annotation method, by using the spatial relationships between objects from images and incorporating such knowledge in a knowledge base, represented as an ontology. Moreover, the semantics of an image may be modeled to increase accuracy of retrieval. The semantics of an image can be reflected by the semantics of relevant objects in the image together with the semantic relationships between them. Thus, a semantic image retrieval system may be designed, which enables users to obtain the semantic clustering of relevant images rather than a list of isolated images as the output of content based image retrieval. Users may then browse images along a semantic path. In this way, the image retrieval system will be able to search and retrieve images in more powerful ways.

REFERENCES

- [1] T. Kato, "Database architecture for content-based image retrieval", Proc. SPIE 1662, vol. 112, pp. 112-123, Feb. 1992. [Online]. Available: <http://dx.doi.org/10.1117/12.58497>
- [2] L. R. B. Schomaker, E. de Leau and L. G. Vuurpijl, "Using pen-based outlines for object-based annotation and image-based queries", Visual Information and Information Systems, pp. 585-592, 1999.
- [3] G. Lu. Multimedia Database Management Systems. Artech House Publishers, Boston, 1999.
- [4] E. Persoon and K. S. Fu, "Shape discrimination using Fourier Descriptors", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 7, pp. 388-397, 1986.
- [5] C. Zahn and R. Roskies, "Fourier descriptors for plane closed curves", IEEE Trans. On Computer, vol. 21, pp. 269-281, 1972.
- [6] D. Zhang and G. Lu, "A comparative study on shape retrieval using fourier descriptors with different shape signatures", Proceedings of International Conference on Intelligent Multimedia and Distance Education, pp. 1-9, 2001.
- [7] E. M. Arkin, L. P. Chew, D. P. Huttenlocher, K. Kedem and J. S. B. Mitchell, "An efficiently computable metric for comparing polygonal shapes", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 13, no. 3, pp. 209-216, 1991. [Online] Available: <http://doi.ieeeecomputersociety.org/10.1109/34.75509>
- [8] K. L. Tan,, B. C. Ooi and L. F. Thiang, "Retrieving similar shapes efficiently", Multimedia Tools and Applications, vol. 19, no. 2, pp. 111-134, 2003.
- [9] S. Fan, "Shape representation and retrieval using distance histograms", unpublished.
- [10] I. Mocanu, "Image retrieval by shape based on contour techniques: a comparative study", Applied Computational Intelligence and Informatics, pp. 219-223, 2007. [Online]. Available: <http://dx.doi.org/10.1109/SACI.2007.375514>
- [11] L.P. Cordella and G. Dettoni, "An O(N) algorithm for polygonal approximation", Pattern Recognition Letters, vol. 3, pp. 93-97, 1985.
- [12] F. Mokhtarian, J. Kittler and S. Abbasi, "The Squid project", unpublished
- [13] I. Mocanu, "Optimization algorithms in multimedia databases", PhD Thesis, unpublished.
- [14] J. P. Eakins, "Automatic image content retrieval – are we getting anywhere?", Proceedings of Third International Conference on Electronic Library and Visual Information Research, pp 123-135, 1996.
- [15] J. P. Eakins, "Techniques for image retrieval", unpublished.
- [16] V. N. Gudivada and V. V. Raghavan, "Content-based image retrieval systems", IEEE Computer, vol. 28, no. 9, pp. 18-22, 1995.
- [17] Y. Mori, H. Takahashi, and R. Oka, "Image-to-word transformation based on dividing and vector quantizing images with words",unpublished.
- [18] P. Duygulu, K. Barnard, J. F. G. de Freitas and D. A. Forsyth, "Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary", European Conference on Computer Vision, pp. 97-112, 2002.
- [19] J. Jeon, V. Lavrenko and R. Manmatha, "Automatic image annotation and retrieval using cross-media relevance models", ACM SIGIR, pp. 119-126, 2003.
- [20] P. Panagi, S. Dasiopoulou, G. Th. Papadopoulos, I. Kompatsiaris and M. G. Strintzis, "A genetic algorithm approach to ontology-driven demantic image analysis", IET International Conference on Visual Information Engineering, pp. 132-137, 2006. [Online]. Available: <http://dx.doi.org/10.1049/cp:20060516>
- [21] S. Jeong, "Histogram-based color image retrieval", unpublished.
- [22] M. J. Swain and D. H. Ballard, D. H., "Color indexing", International Journal of Computer Vision, vol. 7, no. 1, pp 11-32, 1991.
- [23] J. B. MacQueen, "Some methods for classification and analysis of multivariate observations", Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability, vol. 1, pp 281-297, 1967.
- [24] I. Mocanu, E. Kalisz and L. Negreanu, "Genetic algorithms viewed as anticipatory systems", AIP Conference Proceedings, to be published.
- [25] M. Nowostawski and R. Poli, "Parallel genetic algorithm taxonomy", Knowledge-Based Intelligent Information Engineering Systems, pp. 88-92, 1999.