An interactive evolutionary approach for content based image retrieval

Miguel Arevalillo-Herráez, Francesc J. Ferri, Salvador Moreno-Picot
Departament d'Informàtica, Universitat de València
Dr Moliner, 50 46100 Burjassot, Spain
{miguel.arevalillo,francesc.ferri,salvador.moreno}@uv.es

***Abstract*—Content Based Image Retrieval (CBIR) systems aim to provide a means to find pictures in large repositories without using any other information except its contents usually as low-level descriptors. Since these descriptors do not exactly match the high level semantics of the image, assessing perceptual similarity between two pictures using only their feature vectors is not a trivial task. In fact, the ability of a system to induce high level semantic concepts from the feature vector of an image is one of the aspects which most influences its performance. This paper describes a CBIR algorithm which combines relevance feedback, evolutionary computation concepts and ad-hoc strategies in an attempt to fill the existing gap between the high level semantic content of the images and the information provided by the low level descriptors.**

I. INTRODUCTION

Evolutionary computation is concerned about the application of natural selection and genetics concepts to computational models. It embraces a set of techniques based on the processes of mutation, reproduction, competition and selection as genetic algorithms [1], evolution strategies [2], [3] and so on. IR (Information Retrieval) deals with finding digital resources in large databases. In this field, evolutionary computation has been applied e.g. to clustering results [4], query optimization [5] and hybrid simulated annealing-genetic programming relevance feedback [6]. CBIR (Content Based Image Retrieval) systems are a special type of IR system where the information contained in the repository are pictures. CBIR systems that use evolutionary computation have been recently presented in [7], [8], [9].

This paper presents an algorithm which combines interactive evolutionary computation concepts with ad-hoc strategies to produce an efficient relevance feedback algorithm for image retrieval. The system has been implemented and evaluated with very promising results.

II. RELEVANCE FEEDBACK IN IMAGE RETRIEVAL

The purpose of a CBIR system is to allow users to perform searches on image repositories. Since most of the time the user's interests can only be expressed in terms of the semantic meaning of a query, this requires that the system is able to translate these high level semantic concepts into the low level representation of the images. The assumption that semantic similarity is related to the similarity between low level features is implicit to this procedure. Since this assumption does not hold completely true, the objective of most CBIR techniques is to reduce the existing gap between the semantics induced from the low level features and the real high level meaningful semantics of the image.

Relevance feedback originated in the context of traditional text based information retrieval systems [10] and refers to a set of techniques which aim to refine the original query interactively, using feedback information on previously retrieved results provided by the user. In the context of CBIR, they are usually based on some form of supervised learning and allow retrieval systems to reduce the semantic gap, converting the search into an iterative, dynamic and interactive process that gradually adjusts the results to the interest of the user.

The most common form of relevance feedback mechanism consists of requesting the user to judge on the results retrieved at each iteration, either marking the pictures retrieved as "relevant", "not relevant" or "neutral", or by means of a slide bar which assigns a "grade of similarity" to each of the results. This information is then used to recast the query and produce a new set of results, repeating the process until a satisfactory outcome is obtained. With the low amount of information provided at each iteration, and in a limited amount of time imposed by the fact that they are interactive systems, a relevance feedback mechanism aims at minimizing user interaction and fastening the search process.

Older approaches assumed the existence of a query point and an appropriate set of weights in the feature space that if found would provide the answer. In these methods, the relevance feedback mechanism attempts to update the query point so that more emphasis is placed on those features which appear more relevant to the user according to his/her selections [11], [12], [13], [9].

Other approaches for this are Bayesian methods [14], [15], Self Organizing Maps (SOMs) [16] or more recently, Support Vector Machines (SVM) [17], [18] or regression models [19].

III. IMAGE RETRIEVAL AND EVOLUTIONARY COMPUTATION CONCEPTS

An evolutionary algorithm attempts to solve a problem applying Darwin's basic principles of evolution [20] on a population of trial solutions to a problem, called individuals. Genetic algorithms are a class of evolutionary algorithms, which place a special emphasis on the application of genetic operators, such as mutation and crossover. A GA (Genetic Algorithm) uses an encoding method to represent potential solutions to the problem, and a measure that allows a quantitative evaluation of each candidate, called a fitness function.

---

[0] This work has been partially funded by UVEG, FEDER and Spanish MEC through projects UV-AE-20070220, DPI2006-15542-C04-04 and Consolider Ingenio 2010 CSD2007-00018

Starting from an initial population, a GA uses the fitness function to evaluate each candidate. The most promising individuals are allowed to reproduce and determine the next generation of individuals, according to a series of pre-established evolution rules. The process is repeated until a solution is found. The representation of the individuals, and the rules that govern the evolution of the population are key issues in the development of a genetic algorithm. Usually, potential solutions are encoded as binary strings, as sequence of characters or as vectors of integer or floating point numbers, where each element represents a particular aspect of the solution. In genetic terms, the potential solution is called the phenotype, and its representation is referred to as its genotype, genome or chromosome. The evolution rules refer to the parent selection method, and the definition of the mutation and cross-over operations to obtain the offspring genotypes for the next generation individuals. Most parent selection methods are stochastic in order to keep the diversity of the population, preventing premature convergence to a sub-optimal solution. Cross-over operations emulate biological meiosis, and mutations mimic the permanent DNA changes due to transcription errors, radiation, and free chemical radicals which occur in nature.

IGAs are a type of GA which have been successfully applied to many different research areas [21], [22]. When the design of an appropriate fitness function is not possible, IGAs (Interactive Genetic Algorithms) use direct human evaluation. In the context of CBIR, the only truly reliable relevance judgments are those coming directly from the user. It seems then that the use of a fitness function based on human interaction may be an appropriate technique to approach the interactive image retrieval problem.

Perhaps the most related publication to our work is that in [23], where the authors propose the use of an standard IGA using wavelet features. Although they both share the use of interactive evolutionary concepts as part of the underlying principles of the algorithm, the GUI concept, the user feedback mechanism, the genotype representation, the parent selection criteria, the size of the population and the genetic operations are all defined differently, and combined with a series of ad-hoc strategies to accelerate the convergence and improve the results obtained.

IV. THE ALGORITHM

A. Representation

In genetic terms, the feature representation of an image would be the equivalent to its genotype, and the image itself to its phenotype. It is reasonable to assume that the relationship between the genotype and the phenotype is strongly causal [24], and therefore small variations on the genotype space imply small variations on the phenotype space. Since the neighborhood structure under a strongly causal mapping is conserved, the neighbors of the genotype would also be neighbors in the phenotype space.

In principle, the genetic inspired framework presented in this paper can be applied using any set of descriptors which satisfy the condition of strong causality in the genotype-phenotype mapping. In this particular work color information as a feature vector of 30 components each representing a bin on HS (Hue-Saturation) histogram of size 10×3, and texture information as 2 independent feature vectors of 10 components each representing granulometric cumulative distribution functions [25] have been used.

B. Fitness Function

In a genetic algorithm, a fitness function is used to measure the goodness of a solution. At each iteration of the relevance feedback process, the information captured from the user interaction can be used to build this function. We are certain that pictures which have been rated as positive have a phenotype which is closer to that of the query than other images which have been flagged as non relevant or simply not rated. Thus, these are the pictures which should be allowed to reproduce and are assigned a maximum score. Non relevant selections are removed from the population and not allowed to reproduce. It could be argued that the idea of a genetic algorithm would be to assign these a lower fitness, but still give them a small chance to survive. However, pure genetic algorithms may take hundreds of iterations to converge to a solution and we cannot afford such numbers. Therefore, it is necessary to find strategies to accelerate the convergence. The problem with restricting the sets of individuals which are allowed to reproduce to such an extent is that the new generations may lack of the required diversity in the resulting population, getting trapped around a sub-optimal solution. To avoid this we apply an operation that we call population enrichment and helps maintaining population diversity, which consists of introducing some new individuals extracted from the repository. These are the neighbors of the positive selections, and this decision is supported by the strong causality assumed for the genotype-phenotype mapping. Neighbors have similar genotypes, and thus their phenotypes should also be related. Still, this implication shall not be given the same confidence as a positive selection.

Three groups of solution images are formed: the positive selections, images among p-neighbors of a positive selection, and images among the next p-neighbors of a positive selection. A genotype may be present in more than one group and it may appear several times within each group. This strategy avoids getting trapped in a close area of the search space at the same time that preserves population diversity and accelerates the convergence. The value of p depends on the size of the repository. In our implementation, it has been fixed to 20. To calculate the neighbors the distances for each descriptor are computed separately using the Euclidean distance on its vector components, and they are equalized so that all possible values are equally probable. The product rule is then used to produce a composite distance. Since only an approximate ranking and not an exact quantitative evaluation is needed, any consistent measure which offers a reasonable accuracy would be valid. Whilst a very accurate measure may accelerate the convergence of the measure, a less precise metric increases the variety of the population and protects the algorithm against converging to sub-optimal solutions.

Another issue is the treatment of pictures which have been marked as non relevant. A current search history is maintained to keep track of all previous negative selections. For all negative selections, whether from this or a previous iteration, their n-neighborhoods are calculated. The value of n in our implementation is 5. We know that their phenotypes are not desired and the strong causality in the genotype-phenotype mapping is used to induce that their genotypes and their neighbor's are not either. These genotypes should not be allowed to reproduce and thus they are removed from the groups of neighbors of the positive selections, if they were present.

C. *Parents Selection*

Parents are chosen using roulette wheel parent selection. The roulette wheel is initially divided into three sectors, one per each of the groups described in the previous section with associated sizes of 60%, 28% and 12%, respectively. Each of these sectors is further divided between the individuals in a proportional way to the number of appearances of each genotype.

D. *Evolution Rules*

Next, we detail some major pitfalls associated with the direct application of standard cross-over and mutation to our genotype representation:

- The application of cross-over and mutation operations may result in inconsistent genotypes. Features which belong to the same concept may have to satisfy certain restrictions. As an example, bins of a normalized color histogram should add to one, and cumulative functions should be monotonic.
- The mapping between a genotype and a phenotype is not known. Even if the new genotype produced is consistent, its corresponding phenotype needs to be shown to the user for evaluation. Since we do not have a conversion function from the genotype to the phenotype space, some procedure which always produces an existing phenotype is required.
- The number of positive selections may be too small to allow the construction of a consistent new population which ensures population diversity. This is specially true during the first iteration, where just one or two positive selections are common.

To solve these problems, reasonable and genetically consistent ad-hoc strategies are adopted. The first pitfall is attacked considering each group of features as a gen. In GAs, gens are regarded as indivisible units in cross-over operations. Since the gens in the original genotype are all coherent and they are not altered during the process, they will remain consistent. The concept of a gen and its implications in crossover operations are illustrated in figure 1. To avoid that the mutation mechanism alters the structure of the gens, a mutation operation consists of replacing a gen by another chosen at random from the images in the repository.

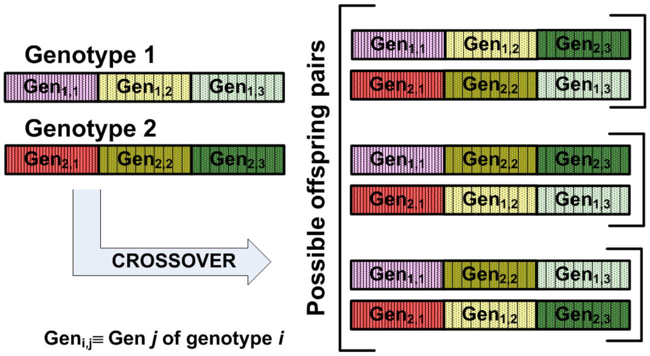

Fig. 1. Possible results of a crossover operation on two genotypes with three gens each.

The second problem is solved searching for the closest genotype in the repository, according to a simple similarity function, and using this instead. This operation can be considered a small mutation of the original genotype to match another genotype which has a known phenotype, and is supported by the strong causality assumed for the genotype-phenotype mapping, which limits the impact of the mutation in the phenotype space.

The solution to the third pitfall has already been described in the context of the fitness function. To form a larger population, we adopted the solution to include new individuals from the neighborhood of the positive selection, and assign a lower fitness value to them.

Each iteration of the algorithm proposed can be described as follows:

1) Repeat k times
    a) Select a pair of genotypes using roulette wheel parent selection.
    b) Apply a cross-over operation to produce a pair of offspring genotypes. In our implementation it is considered that each genotype is composed of three gens. Figure 1 illustrates all possible combinations considered.
    c) Apply a mutation operation on a percentage of the resulting offsprings. In our implementation, this percentage has been fixed to 3%.
    d) Search the pictures which best matches the offspring genotypes in the repository.
    e) Add the genotypes of these pictures to the new population.

Because relevance feedback is an interactive process, the number of times k is a constant defined so that the algorithm executes in an acceptable time. In our implementation, we execute $k = 6000$ matches in each iteration, which allows maintaining the processing time below 2 seconds.

It is worthwhile noticing several aspects of this algorithm:

- The new population may contain the same individual more than once. However, it is not reasonable to display the same image more than once in the same iteration. Instead, a ranking is established so that the individuals that appear more times in the new generation are shown first in the user interface.
- Images which have been flagged as not relevant in previous iterations and appear in the ranking are removed. This avoids that images we already have a valid judgment for, are displayed again.
- Although cross-over operations have been forced to take at least one gen from each parent, the roulette based parent selection algorithm may chose the same individual twice. In this case, the cross-over operation generates two offsprings with the same genotype as the only parent. This produces a desirable effect, causing that the positive user selections be placed at the first ranking positions in the next iterations.
- Mutations have a significant impact in the first iteration of the algorithm, and when the number of positive selections is small. Otherwise, their effect is rather unnoticeable.

E. A Global Perspective

A global schematic view of the algorithm is depicted in Figure 2:

1) First a series of images are displayed on the GUI in an order determined by the ranking established in the previous iteration, and the user flags positive those which are relevant to the query, and negative those which are clearly not relevant.
2) As part of the population enrichment operation, a simple distance function is used to obtain the genotypes of the neighbors of positive and negative selections from the repository. Five sets are created:
    - *Positive selections*. Contains the genotypes of the pictures that the user has flagged as relevant to the query.
    - *Negative selections*. Contains the genotypes of the pictures that the user has flagged as clearly non relevant to the query, at this and any other previous iteration for the same search. A search history is maintained that stores the genotypes which have been rated negative at previous iterations.
    - *Positive first p-neighborhood*. Contains the genotypes of the first p-neighbors of the positive selections according to the distance function used.
    - *Positive next p-neighborhood*. Contains the genotypes of following p-neighbors of the positive selections, according to the same distance function.
    - *Negative n-neighborhood*. Contains the genotypes of the n closest neighbors of the negative selections.
3) Genotypes in the sets *Positive first p-neighborhood* and *Positive next p-neighborhood* which are also included in either set *Negative n-neighborhood* or *Negative selections* are eliminated from their respective sets.

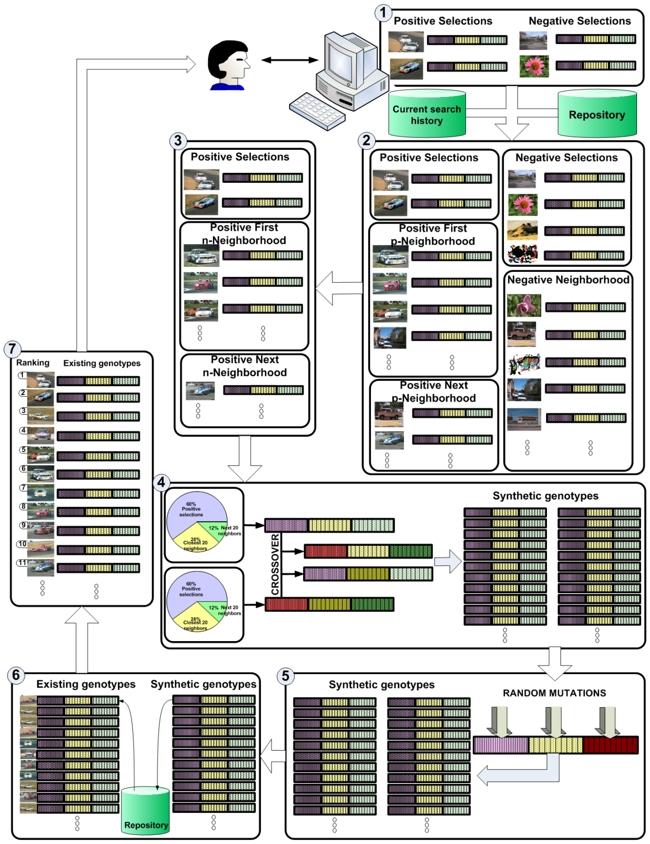

Fig. 2. Schematic description of the entire algorithm

4) The genotypes of the elements in the sets *Positive selections*, *Positive first p-neighborhood* and *Positive next p-neighborhood* are considered representations of potential solutions, and crossover is applied. A roulette based stochastic method is used to select the parents.
5) Random mutations are applied to the resulting set of genotypes. When a mutation occurs, the gen from the synthetic genotype is replaced by the same gen in a picture randomly selected from the repository.
6) The synthetic genotypes are mapped into real genotypes which can be found in the repository.
7) As a result of the previous mapping, some images will appear several times in the new population. These are counted and a ranking is established. This ranking is used to present the new set of images to the user and re-start the process.

V. EVALUATION

The final objective of any CBIR system is that the user is able to find a desired picture in a minimum time. Since time is very sensitive to and highly dependent on many external factors, such as the computer literacy of the user, it is a common approach to use the number of iterations required instead.

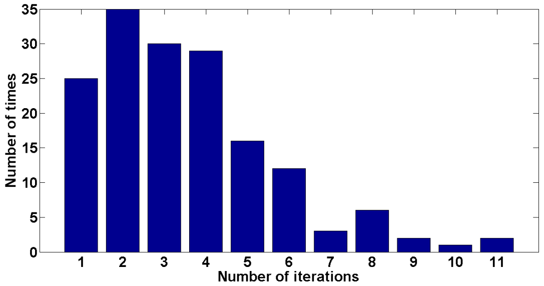

Fig. 3. Histogram of the number of iterations required to bring the query picture into the first page.

Still, the number of iterations depends heavily on two aspects:

1) The picture the user looks for. Some pictures are easier to find that others. If the database contains a large number of pictures which are related to the user query, it will be easier to find.
2) The initial layout of the images. It is easier to find a picture if some images which are related to the query appear on the first few pages of the GUI.

If we are able to run a sufficiently large number of these queries the effects above will minimize. Then, the average number of iterations required to find the target image will become an appropriate measurement of performance.

For evaluation purposes, a special setup that uses the target testing approach [14] has been prepared. The system presents a picture chosen at random from the repository, and the user's aim is to bring that picture into the first page shown. We have used a repository that is composed of 3742 images organized into 49 categories, and we have run a total of 161 successful queries, involving 23 users. A query is considered unsuccessful when a user has run over 8 iterations and the current population does not contain the query image. The evaluation software used saves relevant information into a database. At each iteration, the user is asked to report how many of the results retrieved in the first page are relevant to the query. This value is divided by the number of images in a page -16- to calculate the precision at this cut-off value. Figure 3 shows the number of iterations required as a histogram. An average number of 3.55 iterations were required to find a random target.

To continue with the analysis of the results, we studied the progress of the search process. We have grouped the searches according to the number of iterations which were required to bring the query picture into the first page, and examined the evolution of the precision in those groups where the number of samples makes it possible to obtain a meaningful result. The results are presented in table I.

Each pair of rows in this table represents a set of queries which required the same number of iterations, which is specified in the first column. For each such set, the precision obtained at each iteration is analyzed. This has been calculated at a cut-off value of 16, the number of pictures which are displayed on a page. Both the average and standard deviation of the results obtained for each query in the same group are shown.

A consistent monotonic behavior can be appreciated, an argument in favor of the robustness of the algorithm. However, it is worth noticing some fluctuations which happens for the groups of queries which required over 8 iterations. Although the number of queries in these sets is not statistically meaningful, the non-monotonic behavior deserves a careful analysis.

Intuitively, the images shown at each iteration are within a cloud in the feature space, delimited by the positive selections and their neighbors. The genetic algorithm works moving and re-shaping this cloud towards the images of interest. When the user makes a positive selection which is near the borders of the cloud, the inclusion of the neighbors of the positive selection in the calculation of the new generation causes this effect. Since we are measuring the precision at a cut-off value of 16, we can imagine a cloud of this size. When the images of interest are disperse in the space and the query is located near to a border, it may happen at one stage that the cloud covers a number of related images, but does not include the query itself. As the cloud moves towards the query, some pictures which were previously considered relevant are left outside the scope of the cloud. If these are more than the new relevant images included, the precision value decreases. Obviously this does not mean that the algorithm is diverging, but just the opposite. The larger number of iterations required when this effect has occurred implies a slower convergence, which is consistent with the idea of a cloud slowly moving towards the query. In other cases the cloud is wrapping the query and the effect of user selections is a compression of the cloud, resulting in a faster convergence and requiring less iterations.

TABLE I
AVERAGE OF PRECISION VALUES FOR EACH OF THE 11 GROUPS. THE STANDARD DEVIATION IS ALSO SPECIFIED IN BRACKETS.

Precision values at iteration										
1	2	3	4	5	6	7	8	9	10	11
0.41 (0.22)										
0.23 (0.13)	0.57 (0.23)									
0.27 (0.20)	0.42 (0.20)	0.61 (0.22)								
0.23 (0.14)	0.38 (0.17)	0.51 (0.21)	0.66 (0.18)							
0.26 (0.17)	0.31 (0.19)	0.40 (0.22)	0.42 (0.18)	0.59 (0.22)						
0.19 (0.11)	0.30 (0.13)	0.41 (0.09)	0.48 (0.14)	0.54 (0.16)	0.69 (0.15)					
0.17 (0.16)	0.35 (0.26)	0.33 (0.18)	0.46 (0.19)	0.35 (0.16)	0.46 (0.16)	0.71 (0.14)				
0.21 (0.20)	0.33 (0.18)	0.33 (0.15)	0.43 (0.15)	0.43 (0.13)	0.51 (0.14)	0.51 (0.12)	0.72 (0.13)			
0.28 (0.04)	0.47 (0.13)	0.47 (0.13)	0.47 (0.13)	0.41 (0.04)	0.44 (0.00)	0.41 (0.31)	0.53 (0.31)	0.63 (0.35)		
0.25 (0.00)	0.31 (0.00)	0.31 (0.00)	0.81 (0.00)	0.69 (0.00)	0.25 (0.00)	0.25 (0.00)	0.25 (0.00)	0.25 (0.00)	0.44 (0.00)	
0.13 (0.00)	0.25 (0.00)	0.34 (0.04)	0.41 (0.04)	0.47 (0.04)	0.53 (0.13)	0.59 (0.13)	0.44 (0.09)	0.25 (0.18)	0.25 (0.18)	0.34 (0.22)

VI. DISCUSSION

The algorithm developed in this paper depends heavily of the assumption of strong causality of the genotype-phenotype mapping. This strong causality has made it possible to work with the neighborhoods to enrich the population and map synthetic genotypes to other genotypes which exist in the repository.

In some sense the so-called semantic gap breaks the neighborhood structure under the mapping, and may cause that the algorithm retrieves not only images which are relevant to the query, but also others which are not relevant, simply because they have a similar genotype. In our repository, most stones and flowers share similar representations (genotypes) and therefore they commonly appear together. To clean the space and filter the relevant genotypes we consider negative selections. Removing the neighbors of the negative selections so that they are not allowed to reproduce unless they have been rated positive, has proven to be an effective ad-hoc strategy to palliate this effect.

The genetic technique presented in this work can also be associated with an intuitive reasoning. The system assumes that a user positive selection is based on a strong similarity between some feature of the picture selected and the query. If the algorithm works with color, shape and texture, the system will attempt to discover which one of these features motivated the user selection displaying another series of images, some with the same color as the positive selection, some with the same shape and some with the same texture. The crossover operations on the genotypes make those features which are common to several selections predominant among the new population.

A flexible entire framework which allows the construction of a relevance feedback mechanism for a variety of content based image retrieval systems has been presented. The approach is not only suitable for retrievals based on positive and negative selections, but it can also be adapted to work with other type of paradigms. For example, some engines use a slide bar below each picture which allows the user to assign the image a "grade of similarity", instead of one of three qualifiers. In this case, the grades assigned can be used to alter the probabilities within each group in the roulette, making those pictures with higher grades more likely to reproduce.

REFERENCES

[1] J. H. Holland, *Adaptation in Natural and Artificial Systems*. Cambridge, MA: MIT Press, 1992.

[2] H.-P. Schwefel, *Numerical Optimization of Computer Models*. Chichester: Wiley, 1981.

[3] I. Rechenberg, *Evolutions Strategies*. Stuttgart: Fromman-Holzboog, 1973.

[4] M. Gordon, "User-based document clustering by redescribing subject descriptions with a genetic algorithm," *J Amer Soc Information Sci*, vol. 42, no. 5, pp. 311–322, 1991.

[5] D. Kraft, F. Petry, B. Buckles, and T. Sadasivan, *Genetic algorithms for query optimization in information retrieval: relevance feedback*. World Scientific, 1997, ch. Genetic Algorithms and Fuzzy Logic Systems, p. 155173.

[6] O. Cordón, F. Moya, and C. Zarco, "A new evolutionary algorithm combining simulated annealing and genetic programming for relevance feedback in fuzzy information retrieval systems," *Soft Computing*, vol. 6, pp. 308–319, 2002.

[7] S. Kato, "An image retrieval method based on a genetic algorithm," in *ICOIN '98: Proceedings of the 13th International Conference on Information Networking*. Washington, DC, USA: IEEE Computer Society, 1998, pp. 333–336.

[8] S.-F. Wang, X.-F. Wang, and J. Xue, "An improved interactive genetic algorithm incorporating relevant feedback," in *Proceedings of the 4th International Conference on Machine Learning and Cybernetics*, Guangzhou, August 2005, pp. 2996–3001.

[9] R. da S. Torres, A. X. Falco, M. A. Goncalves, B. Zhang, W. Fan, E. A. Fox, and P. Calado, "A new framework to combine descriptors for content-based image retrieval," in *Fourteenth Conference on Information and Knowledge Management*, Bremen, Germany, 2005, pp. 335–336.

[10] G. Salton and M. J. McGill, *Introduction to Modern Information Retrieval*. McGraw Hill, 1983.

[11] Y. Ishikawa, R. Subramanya, and C. Faloutsos, "Mindreader: Querying databases through multiple examples," in *Proc. 24th Int. Conf. Very Large Data Bases, VLDB*, New York, USA, 1998, p. 433438.

[12] Y. Rui, S. Huang, M. Ortega, and S. Mehrotra, "Relevance feedback: a power tool for interactive content-based image retrieval," *IEEE Transaction on circuits and video technology*, vol. 8, no. 5, 1998.

[13] S. Bissol, P. Mulhem, and Y. Chiaramella, "Dynamic learning of indexing concepts for home image retrieval," in *Content-Based Multimedia Indexing*, Rennes, France, September 2003, pp. 87–93.

[14] I. Cox, M. Miller, T. Minka, T. V. Papathomas, and P. Yianilos, "The bayesian image retrieval system, pichunter: Theory, implementation and psychophysical experiments," *IEEE Transaction on image processing*, vol. 9, no. 1, pp. 20–37, 2000.

[15] E. de Ves, J. Domingo, G. Ayala, and P. Zuccarello, "A novel bayesian framework for relevance feedback in image content-based retrieval systems," *Pattern Recognition*, vol. 39, pp. 1622–1632, 2006.

[16] M. Koskela, J. Laaksonen, and E. Oja, "Use of image subset features in image retrieval with self-organizing maps." in *Image and Video Retrieval: Third International Conference*, Dublin,Ireland, july 2004, pp. 508–516.

[17] X. S. Zhou and T. S. Huang, "Small sample learning during multimedia retrieval using biasmap." in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2001, pp. 11–17.

[18] S. Tong and E. Chang, "Support vector machine active learning for image retrieval," in *Proceedings of the 9th ACM International Conference on Multimedia*. New York, NY, USA: ACM Press, 2001, pp. 107–118.

[19] T. León, P. Zuccarello, G. Ayala, E. de Ves, and J. Domingo, "Applying logistic regression to relevance feedback in image retrieval systems," *Pattern Recogn.*, vol. 40, no. 10, pp. 2621–2632, 2007.

[20] C. Darwin, *The Origin of Species*. London: Dent Gordon, 1973.

[21] H. Takagi, "Interactive evolutionary computation: fusion of the capabilities of ec optimization and human evaluation," *Proceedings of the IEEE*, vol. 89, no. 9, pp. 1275–1296, 2001.

[22] Q.-k. Bu and A.-q. Hu, "Footballs video scene retrieval with interactive genetic algorithm," vol. 2, May 2008, pp. 500–504.

[23] S.-B. Cho and J.-Y. Lee, "A human-oriented image retrieval system using interactive genetic algorithms," *IEEE Transactions on Systems, Man, and Cybernetics. Part A: Systems and humans*, vol. 32, no. 3, pp. 452–458, May 2002.

[24] B. Sendhoff, M. Kreutz, and W. von Seelen, "A condition for the genotype-phenotype mapping: causality," in *Proceedings of the 7th International Conference on Genetic Algorithm*, Bck, Ed. San Francisco: Morgan Kauffman, 1997, pp. 2996–3001.

[25] P. Soille, *Morphological Image Analysis: Principles and Applications*. Berlin: Springer-Verlag, 2003.