INSTITUTO DE COMPUTAÇÃO
UNIVERSIDADE ESTADUAL DE CAMPINAS

A Genetic Programming Approach for Relevance Feedback in Region-based Image Retrieval Systems

J. A. Santos *C. D. Ferreira* *R. S. Torres*

Technical Report - IC-08-19 - Relatório Técnico

August - 2008 - Agosto

A Genetic Programming Approach for Relevance Feedback in Region-based Image Retrieval Systems

Jefersson Alex dos Santos* Cristiano Dalmaschio Ferreira† Ricardo da Silva Torres‡

Abstract

This paper presents a new relevance feedback method for content-based image retrieval using local image features. This method adopts a genetic programming approach to learn user preferences and combine the region similarity values in a query session. Experiments demonstrate that the proposed method yields more effective results than the Local Aggregation Pattern (LAP)-based relevance feedback technique.

1 Introduction

Large image collections have been created and managed in several applications, such as digital libraries, medicine, biodiversity information systems [**?**]. Given the size of these colections, it is essential to provide efficient and effective means to retrieve images.

This is the objective of the so-called *content-based image retrieval (CBIR) systems* [17, 11, 18]. In these systems, the searching process consists in, for a given image, computing the most similar images stored in the database. The searching process relies on the use of image *descriptors*. A descriptor can be characterized by two functions: *feature vector extraction* and *similarity computation*. The feature vectors encode image properties, like color, texture, and shape. Therefore, the similarity between two images is computed as a function of their feature vectors distance.

CBIR approaches can be divided into local and global. In the first one, the feature extraction algorithm considers the whole image in the description process. In the local approaches, the similarity between two images is computed by combining the similarity among their regions.

In fact, for a human, the similarity of images usually is associated with the similarity of objects that can be found in the images. However, for a computer, an object appearing in the image is reduced to a set of pixels, i.e., to an image region. The elaborated characteristics of the human visual system suggest that, in order to approximate the human perception, the image similarity model must capture the region feature properties [21].

---

*Institute of Computing, University of Campinas, 13081-970 Campinas, SP. Research supported by FAPESP — Fundação de Apoio à Pesquisa do Estado de São Paulo and CAPES — Coordenação de Aperfeiçoamento de Pessoal de Nível Superior

†Institute of Computing, University of Campinas, 13081-970 Campinas, SP

‡Institute of Computing, University of Campinas, 13081-970 Campinas, SP.

Motivated by this limitation, *relevance feedback* approaches were incorporated into CBIR systems [19, 4, 12]. This technique makes the user interaction with the retrieval systems possible. Basically, the relevance feedback-based image retrieval process consists in three steps: (i) showing a small number of retrieved images to the user; (ii) indication of relevant or irrelevant images by the user; (iii) finally, learning the user needs from his/her feedback, and selecting a new set of images to be shown. This procedure is repeated until a satisfactory result is reached.

In this paper a new relevance feedback-based CBIR method using region features is proposed. This method adopts a genetic programming approach to learn user preferences and combine the region similarity values in a query session. Genetic programming (GP) [14] is a Machine Learning technique used in many applications, such as data mining, signal processing, and regression [2, 9, 28]. This technique is based on the evolution theory to find optimal solutions. In our method, we aim to find a function that combines the region similarity values computed by different descriptors, and then learn the user needs.

2 Related Works

Relevance feedback (RF) [19, 4, 12, 26] is a technique initially proposed for document retrieval that has been used with great success for human-computer interaction in CBIR. RF addresses two questions referring to CBIR process. The first one is the semantic gap between high-level visual properties of images and low-level features used to describe them. Usually, it is not easy for a user to map his/her visual perception of an image into low level features such as color and shape. Another issue is concerned with the subjectivity of the image perception. Different people can have distinct visual perceptions of the same image. Different images may have different meanings or importance for different users. For example, given a picture showing a "car in front of a house", while a user may be interested in cars, others may be interested in houses.

In [21] the similarity of region features was used by Stejic et al. for image retrieval. They proposed a genetic algorithm (GA)-based relevance feedback method and a new method, Local Similarity Pattern (LSP), for computing image similarity. LSP is defined as a structure containing R and F_R, where R is a set with $N \times N$ regions obtained by the image uniform partitioning, and F_R is a set of image features that are extracted from each region and used for similarity computation. GA and relevance feedback are used to determine the feature that best describes each LSP region. In [23], Stejic et al. proposed the RFSP (Region and Feature Saliency Pattern). The RFSP is defined like a structure such as LSP. But, instead of using GA to determine the feature that best describes each image region, in the RFSP method, there is a weight associated to each region and the GA is used to find the best weights for all region features. A new GA-based relevance feedback technique was proposed in [22]: Local Aggregation Pattern (LAP). In LAP, Stejic et al. used mathematical aggregation operators to combine the similarity regions. So, in this approach, GA-based RF is used to find the best set of mathematical aggregation operators.

There are RF methods based on SVM (*Suport Vector Machine*) using local information. Jing et al. [13] proposed two relevance feedback algorithms based on region representations. One is inspired from the query point of positive examples together and reweinghting the regions to emphatize the latest ones, a pseudo image is formed as the new query. The other propose a new SVM kernel so as enable the algorithms to be applicable to region-based representations. In another approach, Lin et. al. [16] proposed to carry out the recognition task with adaptative ensemble kernel machines, each of which is derived from proper localization and regularization for object category recognition.

Some RF methods that are being applied to CBIR uses image global information. One of the first relevance feedback-based CBIR method was proposed in [19]. Another pioneer work in this area is the *PicHunter* system, presented in [4]. The PicHunter uses a Bayesian framework in the learning process. This mechanism tries to predict the image closer to the user needs. In [7], another approach for *relevance feedback* using Bayesian inference is proposed: the *rich get richer (RGR)*. This method considers the consistency among successive user feedbacks provided in the learning process. In [3], the query pattern is a set of images, instead of a single one. SVM is also a commonly used RF technique based on global image information. In the work proposed in [12], SVM is used as learning method. The experiments showed that a minimum number of positive and negative examples, heuristically chosen as four, is necessary to guarantee the learning. In [26], Tong et al. propose the use of a *support vector machine active learning* method to separate relevant images from the others. On each iteration, the images closer to the separation hyperplane, the most ambiguous ones, are displayed to the user. To the end of the process, the most distant images from hyperplane are shown.

In the aforementioned methods, the learning process is based on either assigning weights to similarity values determined by different descriptors [3, 23] or finding a function to compute the relevance degree of each image [4, 7, 12, 26].

In the method proposed in this paper, the similarity functions defined for all available descriptors are used. Furthermore, the proposed GP framework allows a more complex combination of the similarity values than linear combination. Genetic programming is used to obtain this function.

3 Background

3.1 CBIR model

This paper uses the CBIR model proposed in [6, 5], described in the following.

Definition 1 *An* **image** $\hat{I}$ *is a pair* $(D_I, \vec{I})$*, where:* D_I *is a finite set of* pixels *(points in* $\mathbb{Z}^2$*, that is,* $D_I \subset \mathbb{Z}^2$*), and* $\vec{I} : D_I \rightarrow \mathsf{D}'$ *is a function that assigns to each pixel* p *in* D_I *a vector* $\vec{I}(p)$ *of values in some arbitrary space* D' *(for example,* $\mathsf{D}' = \mathbb{R}^3$ *when a color in the RGB system is assigned to a pixel).*

Definition 2 *A* ***simple descriptor*** *(briefly,* **descriptor***)* D *is defined as a pair* (ϵ_D, δ_D)*, where:* $\epsilon_D : \hat{I} \rightarrow \mathbb{R}^n$ *is a function, which extracts a* feature vector $\vec{v}_{\hat{I}}$ *from an* image $\hat{I}$*.* $\delta_D : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$ *is a* similarity function *(e.g., based on a distance metric) that computes the similarity between two images as the inverse of the distance between their corresponding* feature vectors.

Definition 3 *A* ***feature vector*** $\vec{v}_{\hat{I}}$ *of an image* $\hat{I}$ *is a point in* $\mathbb{R}^n$ *space:* $\vec{v}_{\hat{I}} = (v_1, v_2, ..., v_n)$*, where* n *is the dimension of the vector. They essentially encode image properties, such as color, shape, and texture. Note that different types of feature vectors may require different similarity functions.*

Figure 1a illustrates the use of a simple descriptor D to compute the similarity between two images $\hat{I}_A$ and $\hat{I}_B$. First, the extraction algorithm ϵ_D is used to compute the feature vectors $\vec{v}_{\hat{I}_A}$ and

$\vec{v}_{\hat{I}_B}$ associated with the images. Next, the similarity function δ_D is used to determine the similarity value d between the images.

Definition 4 *A* ***composite descriptor*** $\hat{D}$ *is a pair* $(\mathcal{D}, \delta_{\mathcal{D}})$ *(see Figure 1b), where:* $\mathcal{D} = \{D_1, D_2, \ldots, D_k\}$ *is a set of* k *pre-defined simple descriptors.* $\delta_{\mathcal{D}}$ *is a similarity combination function which combines the similarity values* d_i *obtained from each descriptor* $D_i \in \mathcal{D}$*,* $i = 1, 2, \ldots, k$*.*

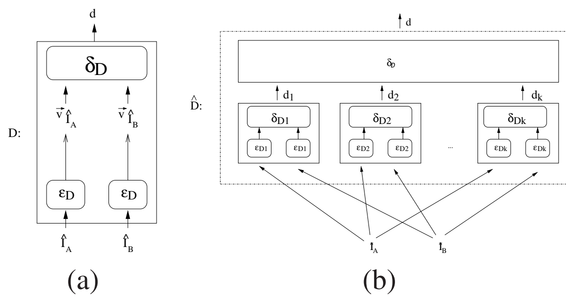

Figure 1: (a) Simple and (b) Composite descriptors.

3.2 Region-Based Image Similarity Model

This paper proposes a RF approach based on local imagem feartures. In the following, a Region-based Image Similarity Model (RISM) used is described.

In general, RISMs express the image similarity as a combination of region similarities [23]. There are many ways to model image similarity based on regions. Some approaches are based on a segmentation-process step. However, to partition the image into grids with the same size is easier. Segmentation needs more complex algorithms and the commonly used data structures are more difficult to be manipulated.

Thus, in this work the RISM used it is based on Stejic et al. methods [21, 23, 22]. A formalization of RISM is explained in the following.

Let I be a *set of images* that represents the image database. Each image is partitioned into a *set of regions* $R = \{r_1, r_2, ...r_{n_R}\}$. Figure 2 ilustrates the partition of an image into 9 regions. From each region, a set of feature vectors $F_{r_i} = \{f1_{r_i}, f2_{r_i}, ...f_{n_D}\}$ are extracted using a set $\mathcal{D} = \{D_1, D_2, \ldots, D_{n_D}\}$ of n_D descriptors.

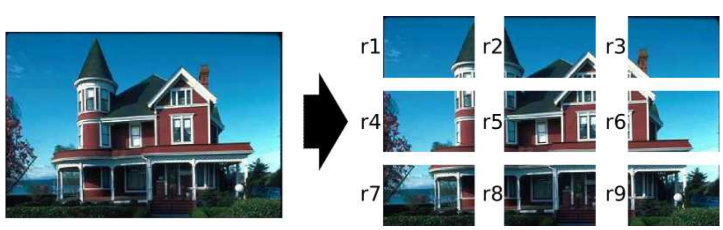

Figure 2: Example of image partition.

A descriptor D_i returns the feature similarity degree $d_i r_{j I_a I_b}$, using a simililarity function δ_{D_i}, from a pair of images I_a and I_b with respect to the image feature f_i of the image region r_j. Given a colection of feature similarity values, a composite descriptor $\mathcal{D}$ returns the image similarity value $d_{i I_a I_b}$ of a pair of images I_a and I_b.

Figure 3 shows the Region-based Image Similarity Model used. For each region of the image, the features vectors and the similarities are calculated by using the k descriptors available. A δ_D function is used to combine the region similarites. It is possible to observe that this structure is a typical composite descriptor like described in Section 3.1.

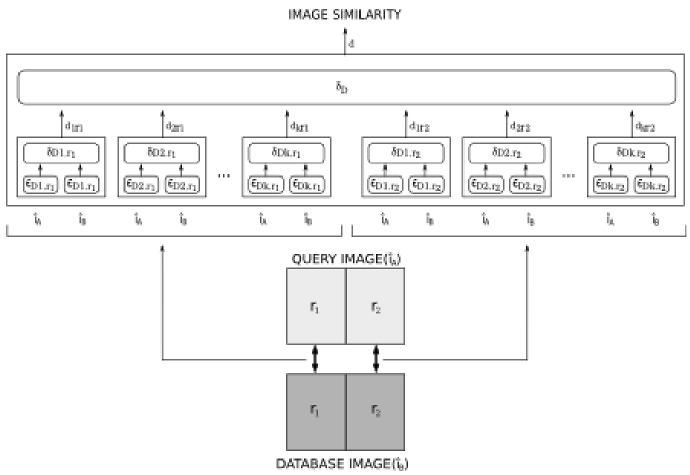

Figure 3: Example of the image similarity computation using the proposed model.

3.3 Genetic Programming

Genetic programming (GP) [14], such as other evolutionary computation algorithms, is an artificial intelligence problem-solving technique based on the principles of biological inheritance and evolution. In GP approach, the individuals represent programs that undergo evolution. The *fitness* evaluation consists in executing these programs, and measuring their degrees of evolution. Genetic programming, then, involves an evolution-directed search in the space of possible computer programs that best solve a given problem.

At beginning of the evolution, an initial population of individuals is created. Next, a loop of successive steps are performed to evolve these individuals: the fitness calculation of each individual, the selection of the individuals, based on their fitness, to breed a new population by applying genetic operators. In the following, these steps are presented in more details.

Usually, a GP individual represents a program and is encoded in a tree. In this enconding, an individual contains two kinds of nodes, *terminals* (leaf nodes) and *functions* (intern nodes). Terminals are usually program inputs, although they may also be constants. Functions take inputs and produce outputs. A function input can be either a terminal or the output of another function.

The fitness of an individual is determined by its effectiveness in producing the correct outputs for all cases in a *training set*. The training set is a set containing inputs and their correspondent previously known outputs.

To evolve the population, and optimize the desired objectives, it is necessary to choose the correct individuals to be subject to genetic operators. Thus, *selection operators* are employed to select the individuals, usually, based on their fitness. Examples of selection method are *roulette wheel*, *tournament* and *rank-based* selections [1].

Genetic operators introduce variability in the individuals and make evolution possible, which may produce better individuals in posterior generations. The *crossover* operator exchanges sub-trees from a pair of individuals, generating two others. *Mutation* operator replaces a randomly chosen sub-tree from an individual by a sub-tree randomly generated. The *reproduction* operator simply copies individuals and insert them in the next generation.

4 Proposed Region-Based Image Similarity Model Using GP

This section presents the proposed GP-based CBIR framework with relevance feedback using image similarity based on regions (GP_{LSP}). In this method, a composite descriptor $\hat{D} = (\mathcal{D}, \delta_{\mathcal{D}})$ (see Section 3.1) is employed to rank N database images defined as $DB = \{db_1, db_2, \ldots, db_N\}$. The set of K simple descriptors of $\hat{D}$ is represented by $\mathcal{D} = \{D_1, D_2, \ldots, D_{n_{\mathcal{D}}}\}$. Database images are partitioned into a set of regions $R = \{r_1, r_2, ...r_{n_R}\}$ (see Section 3.2). The similarity between two image regions $I_{a,rj}$ and $I_{b,rj}$, computed by D_i, is represented by $d_i r_{j I_a I_b}$. All similarities $d_i r_{j I_a I_b}$ are normalized between 0 and 1. A Gaussian normalization [19] can be employed to normalize these values. So, the similarity between two images I_a and I_b are obtained combining the $n_{\mathcal{D}} \times n_R$ image regions similarities.

Let L be a number of images displayed on each iteration. Let Q be the query pattern $Q = \{q_1, q_2, \ldots, q_M\}$, where M is the number of elements in Q, formed by the query image q_1 and all images defined as relevant during a retrieval session.

Algorithm 1 presents an overview of the retrieval process used in this paper. The user interactions are indicated in italic. At the beginning of the retrieval process, the user indicates the query image q_1 (line 1). Based on this image, a initial set of images is selected to be shown to the user (line 2). Thus, the user is able to indicate the relevant images, from this initial set, starting the relevance feedback iterations. Each iteration involves the following steps: user indication of relevant images (line 4); the update of the query pattern (line 5); the learning of the user preference by using GP (line 6); database images ranking (line 7); and the exhibition of the most similar images (line 8).

Algorithm 1 The GP-based relevance feedback process.

1 *User indication of query image* q_1
2 Show the initial set of images
3 **while** the user is not satisfied **do**
4   *User indication of the relevant images*
5   Update query pattern Q
6   Apply GP to find the best individuals (similarity composition functions)
7   Rank the database images
8   Show the L most similar images
9 **end while**

The selection of the initial image set, the use of GP to find the best similarity composition functions and the algorithm to rank database images are presented in details in the following subsections.

4.1 Selecting the initial image set

The initial set of images showed to the user is defined by ranking the database images db_i according to their similarity to the query image q_1. This process is performed in two steps. Firstly, each simple descriptor $D_j \in \mathcal{D}$ is used to compute the similarity $d_{jq_1db_i}$. Next, the arithmetic mean is used to combine all these similarity values, that is $\delta_{MEAN}(q_1, db_i) = \frac{\sum_{j=1}^{K} d_{jq_1db_i}}{K}$.

This combination uses all descriptors available and assigns the same degree of importance to all of them.

Hence, the L first images are exhibited to the user. The user, then, identifies the set $RI = \{RI_1, RI_2, \ldots, RI_P\}$ of P relevant images. All images $\{RI_i | RI_i \notin Q\}$ are inserted into the query pattern Q.

4.2 Finding the best similarity combinations

As aforementioned, the goal of our mechanism is to find the region similarity combination functions that best encode the user needs. We employ GP_{LSP} to find these combinations. As presented in Section 3.3, the GP technique requires the definition of several components, such as selection method, genetic operators, etc. In this section, the individual definition and the fitness computation, are discussed in details.

4.2.1 Individual definition

In our method, each GP individual represents a candidate function $\delta_{\mathcal{D}}$, that is, a similarity combination function. This is encoded in a tree structure, as proposed in [6, 5]. Intern nodes contain arithmetic operators. Leaf nodes have similarities values $d_i r_{jI_aI_b}$, where $1 \leq i \leq K$ and $1 \leq j \leq n_R$. Figure 4 shows an example of an individual. The individual in this figure represents the function $f(d_1r_{1I_aI_b}, d_2r_{1I_aI_b}, d_1r_{2I_aI_b}, d_2r_{2I_aI_b}) = \frac{d_1r_{1I_aI_b} * d_2r_{1I_aI_b}}{d_2r_{2I_aI_b}} + \sqrt{d_1r_{2I_aI_b}}$. This figure considers the use of three distinct descriptors and the set of operators $\{+, /, \log, sqrt\}$ as intern nodes. This individual representation is very suitable for the proposed GP search engine, since it directly encodes the candidates for the $\delta_{\mathcal{D}}$ function.

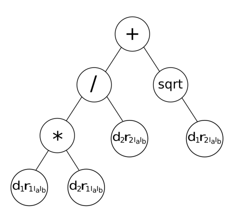

Figure 4: Example of an GP individual.

4.2.2 Individual fitness computation

The goal of the proposed fitness computation process is to assign the highest fitness values to the individuals that best encodes the user preferences. In our approach, the fitness computation is based on the ranking of the database images defined by each individual. Individuals which rank relevant images at the first positions must receive a high fitness value. The proposed fitness computation process is based on this objective criterion.

Training set definition. As aforementioned, the fitness of an individual is computed based on its performance in a training set. In the proposed method, the training set is defined as the following.

Definition 5 *The **training set** is defined as a pair $\mathcal{T} = (T, r)$ where: the* training images *set $T = \{t_1, t_2, \ldots, t_{N_T}\}$ is a set of N_T distinct images. $r : T \rightarrow \mathbb{R}$ is a function that indicates the user feedback for each image in T.*

For instance, $r(t_i)$, where $t_i \in T$, can be defined as

$$r(t_i) = \begin{cases} 1, & \text{if } t_i \text{ is relevant.} \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

It is important that N_T is small ($N_T << N$) to allow a fast computation of each individual fitness. On the other hand, T must also contain a number of images sufficient to represent both, the entire database and the user needs, allowing a suitable evaluation of the individuals. In our approach, T is composed by the last L images exhibited to the user and other $N_T - L$, randomly chosen from the database. To answer the computation time issue, experiments (see Section 5) show that values between 0.5% to 5% of the database size can be suitable choices for N_T. Note that this training set composition represents the user needs, by considering the last exhibited images, and the entire database, by randomly choosing database images.

Fitness computation. The fitness of an individual δ_i is computed based on the similarity between the query pattern and all images from the training set. The fitness computation process is divided into three phases. On the first phase, M ranked lists are computed, each one considering the similarity, according to δ_i, among all training set images and each image in the query pattern. On the second phase, these rankings are evaluated. Finally, on the third phase, the final individual fitness is computed.

Phase 1. On the first phase, for each query pattern image q_j, the training images $t_k \in T$ are sorted according to their similarity ($\delta_i(q_j, t_k)$). The L first images define a ranked list $rk_{j\delta_i}$. Thus, M ranked lists are computed, each one with regard a query pattern image q_j.

Phase 2. Once these rankings are obtained, the second phase starts. The goal of this phase is to evaluate each single ranked list $rk_{j\delta_i}$ generated in Phase 1. This evaluation consists in assigning high values to ranked lists, in which relevant images present in the training set are ranked in the first positions. This evaluation is accomplished by applying an evaluation function $f(rk_{j\delta_i})$ that considers the rank position of the relevant images in $rk_{j\delta_i}$.

In our approach the function $f(rk_{j\delta_i})$ follows the *utility theory* principles [10]. According to this theory, there is an *utility function* which assigns a value to an item, regarding user preference. Usually, it assumes that the utility of an item decreases according it position in a certain ranking [8]. Formally, given two items It_i and It_{i+1}, where i is a position in a ranking, the following condition must be satisfied by a utility function $U(x)$: $U(It_i) > U(It_{i+1})$. In this paper, each item is an image.

An example of $f(rk_{j\delta_i})$ evaluation function is

$$f(rk_{j\delta_i}) = \sum_{l=1}^{L} r(rk_{j\delta_i}[l]) \times k_1 \times \log_{10}(1000/l) \text{ [8]} \quad (2)$$

where k_1 is a constant experimentally defined in [8] as 2, $rk_{j\delta_i}[l]$ is the l^{th} image in the ranking $rk_{j\delta_i}$ and $r(rk_{j\delta_i}) = 1$, if $rk_{j\delta_i}[l] \in R$ or $r(rk_{j\delta_i}) = 0$ otherwise.

Hence, applying $f(rk_{j\delta_i})$ to each ranking $rk_{j\delta_i}$ defines M values $f_{1\delta_i}, f_{2\delta_i}, \ldots, f_{M\delta_i}$.

Phase 3. On the third phase, the final fitness F_{δ_i} of the individual δ_i is computed as the average of the values $f_{j\delta_i}$, that is $F(f_{1\delta_i}, f_{2\delta_i}, \ldots, f_{M\delta_i}) = \frac{\sum_{j=1}^{M} f_{j\delta_i}}{M}$.

4.3 Ranking database images

Once computed the fitness of the individuals, it is possible to define the best individual that will be used to rank the database images. However, it is possible that more than one individual has a high fitness. Actually, if the query pattern size M is small, there is a highly probability that many individuals have a good fitness. Our strategy tries to improve the database images ranking by combining the ranked lists obtained from these “good” individuals. This combination is achieved by applying a *voting scheme*. Let δ_{best} be the best individual obtained from GP (see Section 4.2) in the current iteration. The set S of individuals selected to vote is defined as $S = \{\delta_i | \frac{F_{\delta_i}}{F_{\delta_{best}}} \geq \alpha\}$ where $\alpha \in [0, 1]$ (e.g., $\alpha = 0.95$). The α value is called *voting selection ratio threshold*.

In the voting scheme, all selected individuals vote for β (e.g., $\beta = L$) candidate images. The most voted images are showed to the user.

Firstly, the database images are sorted by using each selected individual δ_i, regarding the similarity $Sim_{\delta_i}(Q, db_j)$, between each image db_j and the query pattern Q. Thus, there is a ranking of images associated to each selected individual $\delta_i \in S$. The similarity function $Sim_{\delta_i}(Q, db_j)$ is defined as the greatest value among $\{\delta_i(q_k, db_j) | 1 \leq k \leq M\}$.

Observe that all images of the query pattern are used to rank the database. Therefore, our approach considers that not only the database images similar to the query image are good candidates to be relevant to the user, but also, those similar to any image belonging to the query pattern.

Each image on the first β (e.g., $\beta = L$) positions in each ranking receives a vote inversely proportional to its position. For instance, the first image receives a vote equal to 1, the second, $1/2$, the third, $1/3$ and so on. Then, the database images are sorted according to the sum of their votes. Finally, the L most voted images are selected to be shown to the user.

5 Experiments

5.1 Image descriptors

The proposed method was presented in a generic way, since there is no restrictions about descriptors that can be used to characterize the images. Color and texture based descriptors are the commonest ones and were used in the experiments. Table 1 shows the used descriptors.

Table 1: Descriptors used in the experiments.

Descriptor	**Sim. Function**	**Type**
Color Histogram [25]	$L1$	Color
Color Moments [24]	d_{mom} [24]	Color
BIC [20]	$dLog$ [20]	Color
Gabor Wavelets [15]	Euclidean	Texture
Spline Wavelets [27]	Euclidean	Texture

5.2 Baselines

We compare our method against the LAP approach proposed by Stejic et al. [22]. As aforementioned in section 2, LAP [22] is a method to compute image similarity based on the similarity of the regions.

LAP computes the similarity of two regions based on local information. This process is comprised of two steps: first, the similarity of regions are computed; second, the region similarity values are combined by means of *mathematical aggregation operators*.

Stejic et al. [22] defined a mathematical aggregation operator a as a function of the form $a : [0,1]^n \mapsto [0,1]$. They used genetic algorithms to find a good set of operators to combine the similality values. The complete set is composed by 67 aggregation operators.

5.3 Image Database

The Image Database used in the experiments was a subset of the heterogeneous colection of 20000 images from the Corel GALLERY Magic — Stock Photo Library 2. The used subset is composed of 3906 images, distributed among 85 classes. These classes have different sizes varying between 7 and 98.

5.4 GP_{LSP} Implementation

We implement a CBIR system with the minimal requirements to validate our method. The configuration parameters used in framework implementation are shown in Table 2.

These parameters were determined empirically through several experiments. As can be seen in this table, only crossover and mutation operators were used in search process. Both uses 2-tournament as selection method. Due to the small population size, the use of reproduction operator makes the population diversity fall down quickly. Thus, this operator was not employed. The protected division used in the function set returns 1 if divisor value was zero. The maximum number of generations adopted was 10, but if a individual has normalized fitness value (between 0 and 1) equal to 1 before the last generation, the GP run is finished earlier. The used fitness function – FFP2 (Equation 2) – is presented in [8].

5.5 Effectiveness measures

We use *precision-recall* curves to evaluate performance in the experiments. Precision-Recall curve is a common performance evaluation criterion used in information retrieval systems that have been

Table 2: Configuration parameters.

Population size	30
Maximum number of generations	10
Maximum tree depth	6
Function set	$+, \times, /$ (*protected*)
Terminal set	similarity by simple descriptors
Initialization	half and half
Initial ramp	$2-6$
Crossover rate	0.80
Mutation rate	0.20
Selection method	tournament (size 2)
Fitness function	FFP2 (Eq. 2)
Training set size	80 (3.6% of DB)
Voting selection ratio threshold	1.00

employed to evaluate CBIR systems. Precision $Pr(q)$ can be defined as the number of retrieved relevant images $R(q)$ over the total number of retrieved images $N(q)$ for a given query q, that is $Pr(q) = \frac{R(q)}{N(q)}$. Recall $Re(q)$ is the number of retrieved relevant images $R(q)$ over the total number of relevant images $M(q)$ present in the database for a given query q, that is $Re(q) = \frac{R(q)}{M(q)}$.

5.6 Experiment design

The user behavior was simulated by computer. At each iteration, all images belonging to the same class of the query are labeled as relevant. Experiments considered 10 iterations for each query. At each iteration, 20 images were displayed. The first set of images displayed, for a given query, are based on the average of the similarity values measured by each employed descriptor. We refer to our approach as GP_{LSP}.

5.7 Results

Figure 5 shows the precision-recall curves of the GP_{LSP} method using different partitionings of the image area with resolution 3×3, 4×4, 5×5, 6×6, and 7×7 regions. The $GP_{LSP(3\times3)}$ presents best results for recall values greater than 0.3.

Figure 6 compares the best GP-Based RF method ($GP_{LSP}(3 \times 3)$) with the LAP method. As can be observed, the proposed method has better effectiveness for all recall values.

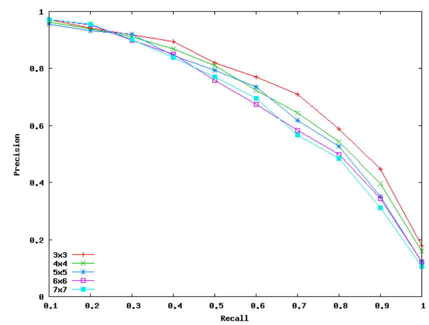

Figure 5: Precision-recall curves showing GP_{LSP} method in 3×3, 4×4, 5×5, 6×6 and 7×7 grids partitions effectiveness in the Corel database.

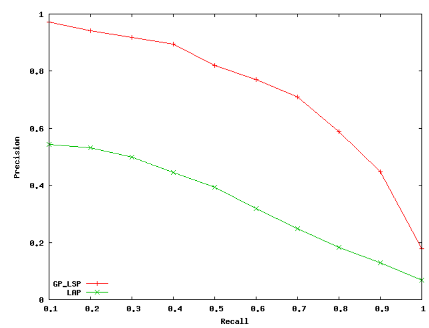

Figure 6: Precision-recall curves showing GP_{LSP} and LAP methods effectiveness in the Corel database.

6 Conclusions

We have presented a genetic programming approach for relevance feedback-based using local properties. This method uses genetic programming to learn the user preferences and combine region similarity features. Experiments showed that the proposed method improves the retrieval effectiveness finding a good composition of descriptors. Moreover, our method outperforms LAP method [22].

One of the next steps of our work is to validate our framework considering other databases. We also plan to compare our method with other relevance feedback techniques [13, 16].

References

[1] T. Bäck, D. B. Fogel, and Z. Michalewicz. *Evolutionary Computation 1 Basics Algorithms and Operators*. Institute of Physics Publishing, 2002.

[2] B. Bhanu and Y. Lin. Object Detection in Multi-Modal Images Using Genetic Programming. *Applied Soft Computing*, 4(2):175–201, May 2004.

[3] M. Cord, J. Fournier, and S. Philipp-Foliguet. Exploration and search-by-similarity in cbir. In *XVI Brazilian Symposium on Computer Graphics and Image Processing*, pages 175–182, 2003.

[4] I. J. Cox, M. L. Miller, T. P. Minka, T. V. Papathomas, and P. N. Yianilos. The Bayesian Image Retrieval System, PicHunter: Theory, Implementation, and Psychophysical Experiments. *IEEE Transactions on Image Processing*, 9(1):20–37, January 2000.

[5] R. da S. Torres, A. X. Falcão, M. A. Gonçalves, B. Zhang, W. Fan, and E. A. Fox. A New Framework to Combine Descriptors for Content-based Image Retrieval. Technical Report IC-05-21, Institute of Computing, State University of Campinas, Campinas, Brazil, 2005.

[6] R. da S. Torres, A. X. Falcão, B. Zhang, W. Fan, E. A. Fox, M. A. Gonçalves, and P. Calado. A new framework to combine descriptors for content-based image retrieval. In *Proceedings of the 14th ACM International Conference on Information and Knowledge Management*, pages 335–336. ACM Press, 2005.

[7] L. Duan, W. Gao, W. Zeng, and D. Zhao. Adaptive relevance feedback based on Bayesian inference for image retrieval. *Signal Processing*, 85(2):395–399, February 2005.

[8] W. Fan, E. A. Fox, P. Pathak, and H. Wu. The Effects of Fitness Functions on Genetic Programming-Based Ranking Discovery for Web Search. *Journal of the American Society for Information Science and Technology*, 55(7):628–636, 2004.

[9] W. Fan, M. D. Gordon, and P. Pathak. A generic ranking function discovery framework by genetic programming for information retrieval. *Information Processing & Management*, 40(4):587–602, July 2004.

[10] P. C. Fishburn. *Non-Linear Preference and Utility Theory*. Johns Hopkins University Press, Baltimore, 1988.

[11] M. Flickner, H. Sawhney, W. Niblack, Q. H. J. Ashley, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker. Query by Image and Video Content: the QBIC System. *IEEE Computer*, 28(9):23–32, Sep 1995.

[12] P. Hong, Q. Tian, and T. S. Huang. Incorporate support vector machines to content-based image retrieval with relevant feedback. In *Proceedings of the 7th IEEE International Conference on Image Processing*, pages 750–753, 2000.

[13] F. Jing, M. Li, H.-J. Zhang, and B. Zhang. Relevance feedback in region-based image retrieval. *Circuits and Systems for Video Technology, IEEE Transactions on*, 14(5):672–681, May 2004.

[14] J. R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, Cambridge, MA, USA, 1992.

[15] T. S. Lee. Image representation using 2d gabor wavelets. *IEEE Transactions Pattern Analysis Machine Intelligence*, 18(10):959–971, 1996.

[16] Y.-Y. Lin, T.-L. Liu, and C.-S. Fuh. Local ensemble kernel learning for object category recognition. *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pages 1–8, 17-22 June 2007.

[17] V. E. Ogle and M. Stonebraker. Chabot: Retrieval from Relational Database of Images. IEEE *Computer*, 28(9):40–48, Sep 1995.

[18] Y. Rui, T. S. Huang, and S. F. Chang. Image Retrieval: Current Techniques, Promising Directions, and Open Issues. *Journal of Communications and Image Representation*, 10(1):39–62, March 1999.

[19] Y. Rui, T. S. Huang, M. Ortega, and S. Mehrotra. Relevance Feedback: A Power Tool for Interactive Content-Based Image Retrieval. *IEEE Transactions on Circuits and Systems for Video Technology*, 8(5):644–655, 1998.

[20] R. Stehling, M. Nascimento, and A. Falcão. A Compact and Efficient Image Retrieval Approach Based on Border/Interior Pixel Classification. In *Proceedings of the 11th ACM International Conference on Information and Knowledge Management*, pages 102–109, McLean, Virginia, USA, November 2002. ACM Press.

[21] Z. Stejic, Y. Takama, and K. Hirota. Genetic algorithms for a family of image similarity models incorporated in the relevance feedback mechanism. *Appl. Soft Comput.*, 2(4):306–327, 2003.

[22] Z. Stejic, Y. Takama, and K. Hirota. Mathematical aggregation operators in image retrieval: effect on retrieval performance and role in relevance feedback. *Signal Processing*, 85(2):297–324, 2005.

[23] Z. Stejic, Y. Takama, and K. Hirota. Relevance feedback-based image retrieval interface incorporating region and feature saliency patterns as visualizable image similarity criteria. *Industrial Electronics, IEEE Transactions on*, 50(5):839–852, Oct. 2003.

[24] M. A. Stricker and M. Orengo. Similarity of Color Images. In *Storage and Retrieval for Image and Video Databases (SPIE)*, pages 381–392, 1995.

[25] M. Swain and D. Ballard. Color Indexing. *International Journal of Computer Vision*, 7(1):11–32, 1991.

[26] S. Tong and E. Y. Chang. Support vector machine active learning for image retrieval. In *Proceedings of 9th ACM international conference on Multimedia*, pages 107–118, New York, NY, USA, 2001. ACM Press.

[27] M. Unser, A. Aldroubi, and M. Eden. A family of polynomial spline wavelet transforms. *Signal Process.*, 30(2):141–162, 1993.

[28] B. Zhang, M. A. Gonçalves, W. Fan, Y. Chen, E. A. Fox, P. Calado, and M. Cristo. Combining structural and citation-based evidence for text classification. In *Proceedings of the 13th ACM Conference on Information and Knowledge Management*, pages 162–163, 2004.