Image Retrieval with Relevance Feedback based on Genetic Programming

Cristiano D. Ferreira[1], Ricardo da S. Torres[1], Marcos André Gonçalves[2], Weiguo Fan[3]

[1]Institute of Computing, University of Campinas – UNICAMP
13083-970, Campinas, SP, Brazil

[2]Department of Computer Science – Federal University of Minas Gerais
Belo Horizonte, MG – Brazil

[3]Virginia Polytechnic Institute and State University
Blacksburg, VA – USA

crferreira@gmail.com,rtorres@ic.unicamp.br,mgoncalv@dcc.ufmg.br,wfan@vt.edu

Abstract. *This paper presents a new content-based image retrieval framework with relevance feedback. This framework employs Genetic Programming to discover a combination of descriptors that better characterizes the user perception of image similarity. Several experiments were conducted to validate the proposed framework. These experiments employed three different image databases and color, shape, and texture descriptors to represent the content of database images. The proposed framework was compared with three other relevance feedback methods regarding their efficiency and effectiveness in image retrieval tasks. Experiment results demonstrate the superiority of the proposed method.*

1. Introduction

Large image collections have been created and managed in several applications, such as digital libraries, medicine, and biodiversity information systems [Torres and Falcão 2006]. Given the large size of these collections, it is essential to provide efficient and effective mechanisms to retrieve images.

This is the objective of the so-called *content-based image retrieval (CBIR) systems* [R. C. Veltkamp 2000]. In these systems, the searching process consists of, for a given query image, finding 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. The similarity between two images is computed as a function of the distance between their feature vectors.

Usually, different descriptors are statically combined [Vadivel et al. 2004, Torres et al. 2008], that is, the descriptor composition is fixed and used to process all queries submitted to the retrieval system. Nevertheless, different people can have distinct visual perceptions of a same image. Thus, a fixed combination of descriptors may not characterize properly this diversity. Furthermore, it is not easy for a user to map her visual perception of an image into low level features such as color and shape ("semantic gap"). Motivated by these limitations, *relevance feedback* (RF) approaches were incorporated into CBIR systems [Rui et al. 1998, Tong and Chang 2001, Cord et al. 2007, Liu et al. 2007].

Basically, the image retrieval process with relevance feedback is comprised of four steps: (i) showing a small number of retrieved images to the user; (ii) user indication of relevant and non-relevant images; (iii) learning the user needs from her feedbacks; (iv) and selecting a new set of images to be shown. This procedure is repeated until a satisfactory result is reached.

An important element of a relevance feedback technique is the learning process. Several relevance feedback methods designed for CBIR systems implement the learning of the user needs by assigning different weights to the descriptors used in the searching process [Rui et al. 1998, Rui and Huang 2000, Doulamis and Doulamis 2006]. This strategy allows only a linear combination of the similarity values defined by each descriptor. However, more complex combination functions may be necessary to express specific user visual perceptions.

Another common drawback of existing RF methods is concerned with the fact that they, in general, ignore the similarity function defined for each available descriptor. On one hand, their learning process is based only on the feature vectors extracted from each database image [Tong and Chang 2001]. On the other hand, these RF techniques define specific distance functions for computing the similarity between two images [Rui and Huang 2000, Doulamis and Doulamis 2006]. In both cases, the overall CBIR system effectiveness may decrease if the similarity functions of the descriptors are not used. In fact, the effectiveness of a descriptor does not depend only on the feature vector codification, but also on the similarity function defined.

In this paper a new relevance feedback method for interactive image search is proposed. This method adopts a genetic programming approach to learn user preferences in a query session. Genetic programming (GP) [Koza 1992] is a kind of evolutionary algorithm which is distinguished from the others mainly by the individual representation. The use of GP is motivated in this work by its capability of exploring large search spaces, and the previous success of using GP in information retrieval [Fan et al. 2004b, de Almeida et al. 2007, Lacerda et al. 2006] and CBIR [Torres et al. 2008] tasks.

The idea of the proposed framework is to use GP to find a function that combines the similarity values computed by different descriptors. On each iteration, the GP-based learning algorithm tries to find a combination function that best represents the user needs. This approach allows a complex combination of the similarity values. Furthermore, the similarity functions defined for each available descriptor are used to compute the overall similarity between two images and the query pattern is composed of multiple query points [Razente et al. 2007].

The effectiveness and efficiency of the proposed method are compared with other relevance feedback techniques for image retrieval tasks. Experiments conducted considering three different image collections and the use of color, texture, and shape descriptors demonstrate that the proposed framework is effective and efficient for CBIR.

2. Related work

Relevance feedback (RF) [Rui et al. 1998, Tong and Chang 2001] is a technique initially proposed for document retrieval that has been used with great success for human-computer interaction in CBIR.

One of the first relevance feedback-based CBIR methods was proposed in [Rui et al. 1998]. In this work, the learning process is based on assigning weights to each descriptor (*interweight*), and also to each feature vector bin, that is, to each position in this vector (*intraweight*). The learning algorithm heuristically estimates the weight values that best encodes the user needs in the retrieval process. This method assumes that the similarity function employed allows the assignment of weights to the feature vector bins.

In [Rui and Huang 2000], the weight assignment is again employed. However, an optimization framework is applied to estimate the weights. This framework is based on the minimization of the Generalized Euclidean distance. Furthermore, this technique uses the *query point movement* approach, which tries to estimate the feature vector of the query pattern that best represents the user perception.

Another learning technique commonly used in RF is *Support Vector Machines (SVM)*. Basically, the goal of SVM-based methods is to find a hyperplane which separates the relevant from the non-relevant images in a high dimensional feature space. In [Tong and Chang 2001], 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. At the end of the process, the most (positive) distant images from hyperplane are shown.

A genetic algorithm (GA)-based relevance feedback method is proposed in [Stejić et al. 2003]. Local similarity pattern (LSP) is used in the retrieval process. 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 the aforementioned methods, the learning process is based on either assigning weights to similarity values determined by different descriptors [Rui et al. 1998, Rui and Huang 2000, Stejić et al. 2003] or finding a function to compute the relevance degree of each image [Rui and Huang 2000, Tong and Chang 2001].

The first group allows only a linear combination of the similarity values. However, a more complex combination among the similarity values may be necessary to express the user needs. The second group ignores the similarity function defined for the used descriptors. Nevertheless, the effectiveness of a descriptor does not depend only on the feature vector codification, but also on the similarity function defined. While the effectiveness of SVM-based methods [Tong and Chang 2001] relies on the discriminatory power of the feature vectors, other approaches define specific similarity functions (e.g., Generalized Euclidean distance [Rui and Huang 2000]) to be employed in the searching process.

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 combination. To the best of our knowledge, even though other kinds of evolutionary algorithms have been used [Stejić et al. 2003], the GP technique was never employed in RF methods.

3. Background

This section presents the CBIR model adopted in our work and a brief overview of the GP basic concepts.

3.1. CBIR model

This paper uses the CBIR model proposed in [Torres and Falcão 2006], described in the following.

Definition 1 *An* **image** $\hat{I}$ *is a pair (*D_I*,* $\vec{I}$*), where 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,* $\mathsf{D}' = \mathrm{I\!R}^3$ *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 by taking into account the distance between their corresponding* feature vectors*.*

Figure 1 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 3 *A* ***composite descriptor*** $\hat{D}$ *is a pair* $(\mathcal{D}, \delta_{\mathcal{D}})$ *(see Figure 2), where:* $\mathcal{D} = \{D_1, D_2, \ldots, D_k\}$ *is a set of* k *predefined simple descriptors.* $\delta_{\mathcal{D}}$ *is a similarity combination function which combines the similarity values* d_i *obtained from each descriptor* $D_i \in \mathcal{D}$*,* $i = 1, 2, \ldots, k$*.*

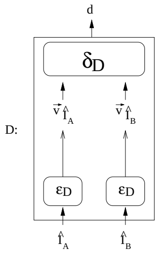


Figure 1. A simple descriptor.

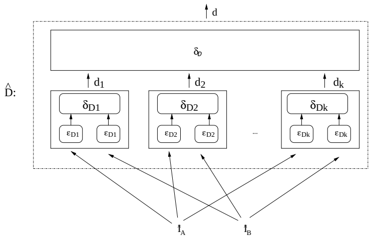


Figure 2. A composite descriptor.

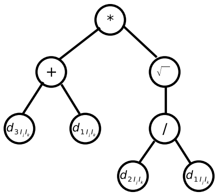


Figure 3. An example of a combination function encoded in a GP individual.

3.2. Genetic Programming

Genetic programming (GP) [Koza 1992], as well as other evolutionary computation algorithms, is an artificial intelligence problem-solving technique based on the principles of biological inheritance and evolution. In the GP approach, the individuals represent programs that undergo evolution. The *fitness* evaluation consists of 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.

Usually, a GP individual represents a program and is encoded in a tree. In this encoding, 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 contains inputs and their previously known correspondent 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.

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. The *mutation* operator replaces a randomly chosen sub-tree from an individual by a sub-tree randomly generated. The *reproduction* operator simply copies individuals and inserts them in the next generation.

4. GP-based RF framework

This section presents the proposed GP-based CBIR framework with relevance feedback, that we call GP^+. 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 $\mathcal{D} = \{D_1, D_2, \ldots, D_K\}$. The similarity between two images I_j and I_k, computed by D_i, is represented by $d_{iI_jI_k}$. All similarities $d_{iI_jI_k}$ are normalized between 0 and 1 [Rui et al. 1998].

Let L be the 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 proposed 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 L most similar images (line 8).

4.1. Selecting the initial image set

The initial set of images showed to the user is defined by ranking the database images according to their similarity to the query image q_1. This process is performed in two

Algorithm 1 The proposed GP-based relevance feedback process.

```
User indication of query image $q_1$
Show the initial set of images
while the user is not satisfied do
    User indication of the relevant images
    Update query pattern $Q$
    Apply GP to find the best individuals (similarity composition functions)
    Rank the database images
    Show the $L$ most similar images
end while
```

steps. First, each simple descriptor $D_j \in \mathcal{D}$ is used to compute the similarity $d_{jq_1db_i}$, where $db_i \in DB$ $(1 \leq i \leq N)$. Next, the arithmetic mean is used to combine all these similarity values, that is $\delta_{MEAN}(q_1, db_i) = \frac{\sum\limits_{j=1}^{K} d_{jq_1db_i}}{K}$.

This combination uses all descriptors available and assigns the same degree of importance to all of them. This kind of combination allows the definition of the initial set of images without previous knowledge about the descriptor effectiveness on the image database used.

Hence, the L first images are exhibited to the user. The user, then, identifies the set $R = \{R_1, R_2, \ldots, R_P\}$ of P relevant images. All images $\{R_i | R_i \notin Q\}$ are inserted into the query pattern Q.

4.2. Finding the best similarity combinations – The GP framework

Only two of our GP framework components require specific definition for the proposed learning process: the individual definition and the fitness computation. The other components used are presented in Section 3.2.

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 [Torres et al. 2008]. Intern nodes contain arithmetic operators. Leaf nodes have similarity values $d_{iI_jI_k}$, where $1 \leq i \leq K$. Figure 3 shows an example of an individual. The individual in this figure represents the function $f(d_{1I_jI_k}, d_{2I_jI_k}, d_{3I_jI_k}) = (d_{1I_jI_k} + d_{3I_jI_k}) * \sqrt{\frac{d_{2I_jI_k}}{d_{1I_jI_k}}}$. This figure considers the use of three distinct descriptors and the set of operators $\{+, /, *, \sqrt{}\}$ as intern nodes.

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 encode the user preferences. In our approach, the fitness computation is based on the ranking of the database images defined by each individual. Individuals that rank relevant images at the first positions must receive the highest fitness values. The proposed fitness computation process is based on this objective criterion.

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

Definition 4 *The **training set** is defined as a pair* $\mathcal{T}^+ = (T, r^+)$ *where* $T = \{t_1, t_2, \ldots, t_{N_T}\}$ *is a set of* N_T *distinct* training images *and* $r^+ : T \to \mathbb{R}$ *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

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

Recall that L images are showed to the user on each iteration. In the GP^+ approach, the training set T is comprised of all M images labeled as relevant until the current iteration, $L - M$ images showed to the user in the last iteration, and $N_T - L$ other images randomly chosen from the database. If $M \geq L$, T is composed of L randomly chosen relevant images and $N_T - L$ randomly chosen from the database. The addition of non-labeled images into the training set aims to create conditions in which individuals evolve to be able to rank unknown database images effectively.

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 one, M ranked list 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 last phase, the final individual fitness is computed.

Phase 1. In 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 to 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 at 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 [Fishburn 1988]. According to this theory, there is an *utility function* which assigns a value to an item, regarding the user preference. Usually, it assumes that the utility of an item decreases according to its position in a certain ranking [Fan et al. 2004a].

The evaluation function $f(rk_{j\delta_i})$ adopted is:

$$f(rk_{j\delta_i}) = \sum_{l=1}^{L} r(rk_{j\delta_i}[l]) \times k_1 \times \log_{10}(1000/l) \text{ [Fan et al. 2004a]} \tag{2}$$

where k_1 is a constant experimentally defined in [Fan et al. 2004a] 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 $rk_{j\delta_i}[l]$ is relevant or $r(rk_{j\delta_i}) = 0$, otherwise. This function was chosen due to good performance on image retrieval tasks, as reported in [Torres et al. 2008]. 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 the fitness of each individual is computed, it is possible to define the best individual that will be used to rank the database images. 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 by using 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 $\alpha \in [0, 1]$. The α value is called *voting selection ratio threshold*.

In the voting scheme, all selected individuals vote for L candidate images. The most voted images are showed to the user. Algorithm 2 presents this process in details.

Algorithm 2 The voting scheme used to rank the database images.

```
 1 Input:  Set $S$ of selected individuals, database $DB$, query pattern $Q$, number of displayed images $L$.
 2 Output:  ranked list of images to be displayed.
 3 for all $\delta_i \in S$ do
 4     for all $db_j \in DB$ do
 5         $rk_i[j].key \leftarrow Sim^{+}_{\delta_i}(Q, db_j)$
 6         $rk_i[j].element \leftarrow db_j$
 7     end for
 8     Sort $rk_i[j]$
 9     for $j \leftarrow 1$ to $\beta$ do
10         $votes[rk_i[j].element] \leftarrow votes[rk_i[j].element] + 1/j$
11     end for
12 end for
13 Sort $DB$ images regarding their $votes$
14 return  the $L$ most voted images
```

Firstly, the database images are sorted by using each selected individual δ_i, regarding the similarity $Sim^{+}_{\delta_i}(Q, db_j)$, between each database image $db_j \in DB$ and the query pattern Q (lines 3–8). This way, each selected individual δ_i defines a ranked list containing the database images.

Given a set of images $IMG = \{img_1, img_2, \ldots, img_n\}$ and an image I, let $max_{\delta_i}(IMG, I)$ be the greatest similarity value between I and each image of the set IMG, as defined in Equation 3.

$$\begin{gathered} max_{\delta_i}(IMG, I) = \{\delta_i(img_k, I) \,|\, \delta_i(img_k, I) > \delta_i(img_l, I) \\ \forall\, img_k, img_l \in IMG \,\wedge\, k \neq l\} \end{gathered} \quad (3)$$

The similarity function $Sim^{+}_{\delta_i}(Q, db_j)$ is defined as $Sim^{+}_{\delta_i}(Q, db_j) = max_{\delta_i}(Q, db_j)$.

Each image at the first L positions in each ranking list receives a vote inversely proportional to its position (lines 9–11). 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 (line 13). Finally, the L most voted images are selected to be shown to the user (line 14).

5. Experiments

This section describes in details the experiments performed to validate our framework.

5.1. Image Descriptors

The proposed framework was presented in a generic way, since there is no restrictions regarding descriptors that can be used to characterize the images. Color, shape, and texture based descriptors are the most common ones and were used in our experiments.

Table 1 shows the image descriptors used in the experiments. In this table, column *Descriptors* refers to the descriptor name; *Type* presents the type of the descriptor (color, texture, or shape); *Dimension* refers to the size of feature vector extracted by each descriptor; and *Distance function* defines the distance function employed by each descriptor.

Table 1. Image descriptors used in our experiments.

Descriptor	Type	Dimension	Distance function
Color Histogram [Swain and Ballard 1991] (**C1**)	Color	256	L1
Color Moments [Stricker and Orengo 1995] (**C2**)	Color	9	d_{mom} [Stricker and Orengo 1995]
BIC [Stehling et al. 2002] (**C3**)	Color	128	L1
Gabor Filters [Lee 1996] (**T1**)	Texture	32	Euclidean
Spline [Unser et al. 1993] (**T2**)	Texture	26	Euclidean
Moment Invariants [Gonzalez and Woods 1992] (**S1**)	Shape	14	Euclidean
Fourier [Gonzalez and Woods 1992] (**S2**)	Shape	126	Euclidean
MS Fractal [Torres et al. 2004] (**S3**)	Shape	25	Euclidean
BAS [Arica and Vural 2003] (**S4**)	Shape	180	OCS [Wang and Pavlids 1990]
SS [Torres and Falcão 2007] (**S5**)	Shape	30	OCS [Wang and Pavlids 1990]

5.2. Baselines

The proposed framework was compared with other three RF methods: WD_{heu} [Rui et al. 1998], WD_{opt} [Rui and Huang 2000], and SVM_{active} [Tong and Chang 2001]. The first two methods [Rui et al. 1998, Rui and Huang 2000] are based on weight assignment, while the last one [Tong and Chang 2001] relies on the use of SVM to classify database images as relevant and non-relevant. These methods [Rui et al. 1998, Rui and Huang 2000, Tong and Chang 2001] are usually used as baselines in experiments to validate RF techniques [Doulamis and Doulamis 2006, Rui and Huang 2000, Cord et al. 2007]. These baseline methods are described in Section 2.

5.3. Image Collections

Table 2 presents the image collections employed in our experiments. The first column refers to the collection names. The second shows the size of each collection, the number of classes and the size of each class. Finally, the third column presents the descriptors employed on each image collection (see Table 1).

Table 2. Image collections used.

Name	Size(#classes/class sizes)	Descriptors
FISH	$11,000(1,100/10)$	S1, S2, S3
MPEG7	$1,400(70/20)$	S1, S2, S3, S4, S5
COREL	$3,906(85/7-98)$	C1, C2, C3, T1, T2

The use of the FISH collection aims to evaluate the effectiveness of the proposed frameworks when the number of relevant images for a given query is very small.

The experiment with MPEG7 aims to evaluate the impact of not using appropriate similarity functions for each available descriptor.

As mentioned in Section 2, WD_{heu} approach requires the use of a similarity function that allows the assignment of weights for each feature vector bin. We use the *Weighted Euclidean Distance* for computing the similarity of BAS and SS feature vectors, since the OCS matching algorithm does not support weight assignment. The WD_{opt} implementation is based on the use of *Generalized Euclidean distance* for computing the similarity between two feature vectors. The SVM_{active} method relies only on the use of feature vectors. No descriptor similarity function is used in the learning process.

The use of COREL collection aims to evaluate the proposed framework with regard to their effectiveness in retrieving colorful images from a heterogeneous collection, where the number of relevant images per class is not balanced (varying from 7 to 98). This collection represents, therefore, a real-world scenario for validating the proposed RF methods.

5.4. Effectiveness Measures

We use two different measures to evaluate the effectiveness of the proposed RF framework: *Precision vs. Recall* curve ($P \times R$) and *Retrieved relevant images vs. number of iterations* curve ($Rel \times It$). A *paired Wilcoxon test* is also used to evaluate the statistical significance of the results.

$P \times R$ curve is a common effectiveness evaluation criterion used in information retrieval systems that have been employed to evaluate CBIR systems. Precision $P(q)$ can be defined as the number of retrieved relevant images $Rel(q)$ over the total number of retrieved images $N(q)$ for a given query q, that is $P(q) = \frac{Rel(q)}{N(q)}$. Recall $R(q)$ is the number of retrieved relevant images $Rel(q)$ over the total number of relevant images $M(q)$ present in the database for a given query q, that is $R(q) = \frac{Rel(q)}{M(q)}$. We use the $P \times R$ curve considering the results obtained on the last RF iteration (e.g., 10th).

$Rel \times It$ curves are used to show the percentage of relevant images retrieved to the user given a number of RF iterations. This curve allows evaluating how the number of retrieved relevant images grows over iterations. For iteration zero, we consider the number of relevant images retrieved in the initial set (see Section 4.1).

The average $P \times R$ and $Rel \times It$ curves, considering the results for all query images are used to compare the RF approaches.

We also performed a paired Wilcoxon test comparing the proposed RF frameworks with all baselines. This test aims to verify if both $P \times R$ and $Rel \times It$ curves obtained by using the proposed framework in CBIR tasks are statistically (significant) different from all the others. We used a Wilcoxon test because the samples do not have a normal distribution.

6. Experiment Results

Two different experiments were conducted: the first one aims to determine the best GP parameters to be used in the RF framework; the second compares the proposed method with the baselines RF techniques with regard to their effectiveness and efficiency.

In our experiments, the presence of users is simulated. In this simulation, all images belonging to the same class of the query image are considered relevant. 10 iterations were considered for each query and 20 images are showed to the user on each iteration.

6.1. GP Parameters

The implementation of the proposed framework requires the definition of several GP parameters. Examples of parameters include population size, maximum number of generations, and genetic operators rate.

We tested the combination of these parameters with regard to their effect in the GP^+ framework. We conducted experiments in the FISH and COREL collections. 550 and 85 randomly chosen images were used as query images for the FISH and COREL collections, respectively.

Table 3 shows the best values found for the GP framework parameters. The parameters found for the COREL collection were also employed in the experiments with the MPEG7 collection.

Table 3. Best parameter values for GP^+.

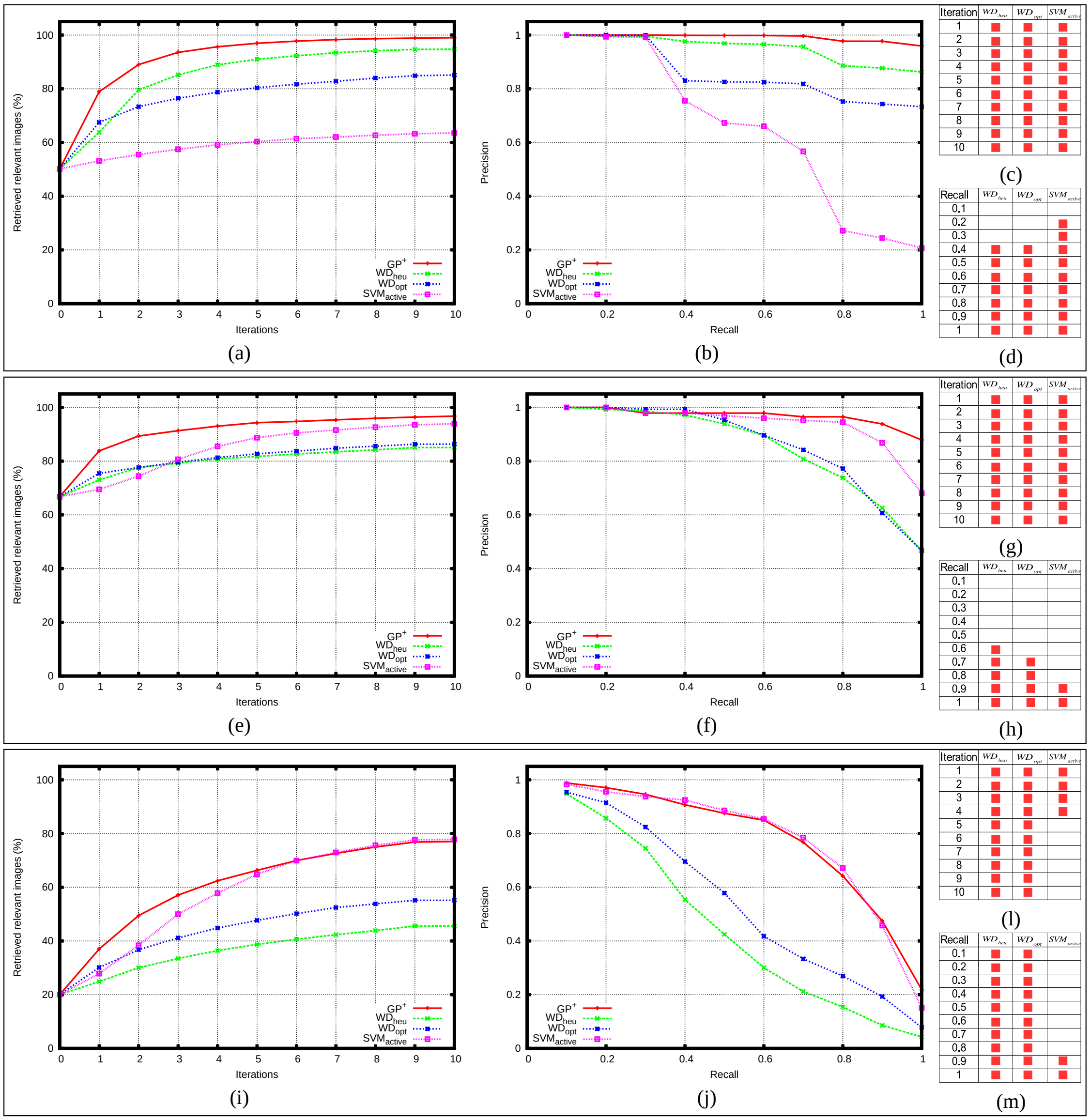
Parameter	FISH Collection	COREL Collection
population size	60	60
number of generations	10	20
initial population	*half-and-half* [Bäck et al. 2002]	*half-and-half* [Bäck et al. 2002]
initial tree depth	$2-5$	$2-5$
maximum tree depth	15	5
selection method	tournament (size 2)	tournament (size 2)
crossover rate	0.8	0.8
mutation rate	0.2	0.2
training set size	55	80
voting selection ratio threshold	1	0.999
functions set	$+, *, /(protected), \sqrt{}$	$+, *, /(protected), \sqrt{}$

6.2. Comparison with Baselines

Experiments aiming to compare the proposed framework with the baselines were conducted using the three data sets presented in Section 5.3. For the FISH collection, 1 image of each class was randomly chosen as query image. Therefore, experiments using this collection considered 1100 query images. 140 query images were used for the MPEG7 collection, which represents 2 randomly chosen from each available class. For the COREL collection, we also used 2 images per class, which represents 170 query images. Experiments were conducted on a 3.0GHz Pentium 4 with 2GB RAM.

Figure 4 shows the experimental results for the FISH, MPEG7, and COREL collections, respectively. That figure presents the *Retrieved relevant images vs. number of iterations* ($Rel \times It$) curves, the *Precision vs. Recall* ($P \times R$) curves and tables with

Wilcoxon significance test. In the Wilcoxon test tables each cell indicates if there is significance statistical difference between the proposed framework – GP^+ (squares) – and each baseline (WD_{heu}, WD_{opt}, and SVM_{active}) for a given recall/iteration value. The presence of squares indicates statistical difference – significance level (α) less than 0.05.



Iteration	WD_{heu}	WD_{opt}	SVM_{active}
1	🟥	🟥	🟥
2	🟥	🟥	🟥
3	🟥	🟥	🟥
4	🟥	🟥	🟥
5	🟥	🟥	🟥
6	🟥	🟥	🟥
7	🟥	🟥	🟥
8	🟥	🟥	🟥
9	🟥	🟥	🟥
10	🟥	🟥	🟥

(c)

Recall	WD_{heu}	WD_{opt}	SVM_{active}
0.1			
0.2			🟥
0.3			🟥
0.4	🟥	🟥	🟥
0.5	🟥	🟥	🟥
0.6	🟥	🟥	🟥
0.7	🟥	🟥	🟥
0.8	🟥	🟥	🟥
0.9	🟥	🟥	🟥
1	🟥	🟥	🟥

(d)

Iteration	WD_{heu}	WD_{opt}	SVM_{active}
1	🟥	🟥	🟥
2	🟥	🟥	🟥
3	🟥	🟥	🟥
4	🟥	🟥	🟥
5	🟥	🟥	🟥
6	🟥	🟥	🟥
7	🟥	🟥	🟥
8	🟥	🟥	🟥
9	🟥	🟥	🟥
10	🟥	🟥	🟥

(g)

Recall	WD_{heu}	WD_{opt}	SVM_{active}
0.1			
0.2			
0.3			
0.4			
0.5			
0.6	🟥		
0.7	🟥	🟥	
0.8	🟥	🟥	
0.9	🟥	🟥	🟥
1	🟥	🟥	🟥

(h)

Iteration	WD_{heu}	WD_{opt}	SVM_{active}
1	🟥	🟥	🟥
2	🟥	🟥	🟥
3	🟥	🟥	🟥
4	🟥	🟥	🟥
5	🟥	🟥	
6	🟥	🟥	
7	🟥	🟥	
8	🟥	🟥	
9	🟥	🟥	
10	🟥	🟥	

(l)

Recall	WD_{heu}	WD_{opt}	SVM_{active}
0.1	🟥	🟥	
0.2	🟥	🟥	
0.3	🟥	🟥	
0.4	🟥	🟥	
0.5	🟥	🟥	
0.6	🟥	🟥	
0.7	🟥	🟥	
0.8	🟥	🟥	
0.9	🟥	🟥	🟥
1	🟥	🟥	🟥

(m)

Figure 4. Experiment results: (a) $Rel \times It$ and (b) $P \times R$ curves from FISH collection; (c) and (d) Wilcoxon test for the data in Figures 4(a) and (b), respectively; (e) $Rel \times It$ and (f) $P \times R$ curves from MPEG7 collection; (g) and (h) Wilcoxon test for the data in Figures 4(e) and (f), respectively; (i) $Rel \times It$ and (j) $P \times R$ curves from COREL collection; (l) and (m) Wilcoxon test for the data in Figures 4(i) and (j), respectively.

As can be observed in Figure 4(a), GP^+ framework presents the best result from the first iteration on for the FISH collection. Note that, from the eighth iteration on, the number of retrieved images is close to 100%, that is, almost all relevant images are returned for all queries. Another important remark is concerned with the bad effectiveness behavior of the SVM_{active} approach. We believe that this result is due to the fact that SVM was not able to learn from the small number of relevant images (in this case only 10) for each query image.

From the Wilcoxon tests (Figure 4(c)), it can be observed that the difference of the $Rel \times It$ curves are significant for almost all iteration values.

The superiority of the proposed framework is also confirmed by the $P \times R$ curves on the last iteration (Figure 4(b)) and by the Wilcoxon test results (Figure 4(d)), considering recall values greater than 0.5.

Also in MPEG7 collection, the $Rel \times It$ curves (Figure 4(e)) show that the number of relevant retrieved images for GP^+ is higher than all baselines. This was confirmed by the Wilcoxon test (Figures 4(g)). For $P \times R$ curves (Figure 4(f)), all evaluated methods present similar results until a recall value equal to 0.6. From this point on, the GP^+ method starts yielding significant better results (Figure 4(h)).

Considering $Rel \times It$ curves (Figure 4(i)) on COREL collection, our method and SVM_{active} present better results than other ones. In fact, SVM_{active} is close to GP^+ from the fifth iteration on (Figure 4(i)). The Wilcoxon tests (Figure 4(l)) confirm this observation.

The proposed framework and SVM_{active} method also present better results than other techniques in the $P \times R$ curves on the last iteration (Figure 4(j)). Similarly to the results in Figure 4(i), SVM_{active} is close to GP^+ effectiveness. The Wilcoxon test (Figure 4(m) shows that the results are statistically different from recall value 0.2 on, with regard to WD_{heu} and WD_{opt}. Regarding SVM_{active}, this test shows that for recall values greater than 0.9 GP^+ yields better results.

6.3. Performance Evaluation

One of the key aspects that need to be addressed during the definition of RF techniques is concerned with the real time execution requirements [Zhou and Huang 2003].

Table 4 shows the average execution time on each RF iteration for all evaluated RF approaches and for all three collections used in our experiments. The presented time refers to the average time required by each RF approach to learn the user needs and to select collection images to be showed on each iteration.

Table 4. Average execution time (in seconds).

Base	GP^+	WD_{heu}	WD_{opt}	SVM_{active}
FISH	0.28	0.08	0.81	12.8
MPEG7	0.84	0.02	0.40	5.34
COREL	1.75	0.07	4.43	15.04

As can be observed in Table 4, WD_{heu} yields the best results for all collections. However, regarding its effectiveness, this RF approach is worse than the proposed GP-based framework for all recall/iteration values (see Figure 4). A similar behavior can be observed for WD_{opt}. Even though this approach is faster than GP^+ for some collections its overall effectiveness is worse.

SVM_{active} presents the best effectiveness results among the baselines, but for the FISH collection. For the MPEG7 and COREL collections, for examples, its effectiveness is closer to the proposed framework. However, this method is at least six times slower than GP^+ (see Table 4).

The execution time (between 0.28 and 1.75 seconds) of the proposed framework is acceptable. This performance is dependent on the choice of the GP parameters (see Section 6.1). GP^+ were able to learn the user perception after a few generations, using small population sizes and low trees.

7. Conclusions

We have presented a novel relevance feedback-based CBIR framework. This method uses genetic programming to learn the user preferences, using the similarity functions defined for all available descriptors. The objective of the GP-based learning method is to find a descriptor combination function that best represents the user perception.

Experiments were performed on three different image collections using several descriptors to characterize color, texture, and shape features. In these experiments, the proposed method was compared with three other relevance feedback techniques [Rui et al. 1998, Rui and Huang 2000, Tong and Chang 2001], regarding their effectiveness and efficiency in image retrieval tasks. Furthermore, statistical significance tests were performed to compare experimental results. Our framework presents the best results in terms of effectiveness methods with acceptable response time.

Future work includes the extension of the proposed framework to support the definition of degrees of relevance associated to each image showed to the user. Another important issue that will be investigated is the use of non-relevant images in the learning process.

Acknowledgment

The authors would like to thank CAPES, FAPESP, CNPq, Microsoft Research, and FAPEMIG for the financial support.

References

Arica, N. and Vural, F. T. Y. (2003). BAS: A Perceptual Shape Descriptor Based on the Beam Angle Statistics. *Pattern Recognition Letters*, 24(9-10):1627–1639.

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

Cord, M., Gosselin, P. H., and Philipp-Foliguet, S. (2007). Stochastic exploration and active learning for image retrieval. *IVC*, 25(1):14–23.

de Almeida, H. M., Gonçalves, M. A., C., M., and Calado, P. (2007). A Combined Component Approach for Finding Collection-adapted Ranking Functions based on Genetic Programming. In *SIGIR'07*, pages 399–406, Amsterdam, The Netherlands.

Doulamis, N. and Doulamis, A. (2006). Evaluation of relevance feedback schemes in content-based in retrieval systems. *Signal Processing: Image Communication*, 21(4):334–357.

Fan, W., Fox, E. A., Pathak, P., and Wu, H. (2004a). 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.

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

Fishburn, P. C. (1988). *Non-Linear Preference and Utility Theory*. Johns Hopkins University Press, Baltimore.

Gonzalez, R. C. and Woods, R. E. (1992). *Digital Image Processing*. Addison-Wesley, Reading, MA, USA.

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

Lacerda, A., Cristo, M., Gonçalves, M. A., Fan, W., Ziviani, N., and Ribeiro-Neto, B. (2006). Learning to advertise. In *SIGIR'06*, pages 549–556.

Lee, T. S. (1996). Image representation using 2d gabor wavelets. *IEEE TPAMI*, 18(10):959–971.

Liu, Y., Zhang, D., Lu, G., and Ma, W.-Y. (2007). A survey of content-based image retrieval with high-level semantics. *PR*, 40(1):262–282.

R. C. Veltkamp, M. T. (2000). Content-based image retrieval systems: A survey. Technical report, UU-CS-2000-34.

Razente, H., Barioni, M. C. N., Traina, A. J. M., and Jr., C. T. (2007). Constrained aggregate similarity queries in metric spaces. In *SBBD 2007*, pages 145–159.

Rui, Y. and Huang, T. (2000). Optimizing learning in image retrieval. In *Proc. of the IEEE Conf. on CVPR*, pages 236–245.

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

Stehling, R., Nascimento, M., and Falcão, A. (2002). A Compact and Efficient Image Retrieval Approach Based on Border/Interior Pixel Classification. In *CIKM'02*, pages 102–109.

Stejić, Z., Takama, Y., and K. (2003). Genetic algorithm-based relevance feedback for image retrieval using local similarity patterns. *Information Processing and Management*, 39(1):1–23.

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

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

Tong, S. and Chang, E. Y. (2001). Support vector machine active learning for image retrieval. In *Proc. of 9th ACM inter. conf. on Multimedia*, pages 107–118.

Torres, R. and Falcão, A. X. (2006). Content-Based Image Retrieval: Theory and Applications. *Revista de Informática Teórica e Aplicada*, 13(2):161–185.

Torres, R. and Falcão, A. X. (2007). Contour Salience Descriptors for Effective Image Retrieval and Analysis. *Image and Vision Computing*, 25(1):3–13.

Torres, R., Falcão, A. X., and da F. Costa, L. (2004). A Graph-based Approach for Multiscale Shape Analysis. *Pattern Recognition*, 37(6):1163–1174.

Torres, R., Falcão, A. X., Gonçalves, M. A., Papa, J. P., Zhang, B., Fan, W., and Fox, E. A. (2008). A genetic programming framework for content-based image retrieval. *Pattern Recognition*. to appear.

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

Vadivel, A., Majumdar, A., and Sural, S. (2004). Characteristics of weighted feature vector in content-based image retrieval applications. *Intelligent Sensing and Information Processing*, 1(18):127–132.

Wang, Y. P. and Pavlids, T. (1990). Optimal Correspondence of String Subsequences. *IEEE TPAMI*, 12(11):1080–1087.

Zhou, X. S. and Huang, T. S. (2003). Relevance feedback in image retrieval: A comprehensive review. *Multimedia System*, 8(6):536–544.