Feature Space Optimization for Content-Based Image Retrieval

Letricia P. S. Avalhais, Sergio F. da Silva,
Jose F. Rodrigues Jr., Agma J. M. Traina and
Caetano Traina Jr.
{letricia, sergio, junio, agma, caetano}@icmc.usp.br
Institute of Mathematics and Computer Science
University of São Paulo
São Carlos, Brazil

ABSTRACT

Substantial benefits can be gained from effective Relevance Feedback techniques in content-based image retrieval. However, existing techniques are limited due to computational cost and/or by being restricted to linear transformations on the data. In this study we analyze the role of nonlinear transformations in relevance feedback. We present two promising Relevance Feedback methods based on Genetic Algorithms used to enhance the performance on the task of image retrieval according to the user's interests. The first method adjusts the dissimilarity function by using weighting functions while the second method redefines the features space by means of linear and nonlinear transformation functions. Experimental results on real data sets demonstrate that our methods are effective and the results show that the transformation approach outperforms the weighting approach, achieving a precision gain of up to 70%. Our results indicate that nonlinear transformations have a great potential in capturing the user's interests in image retrieval and should be further analyzed employing other learning/optimization mechanisms[1].

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Search Process, Relevance Feedback.

General Terms

Algorithm, Experimentation.

Keywords

Image Retrieval, Genetic Algorithm, Weighting, Functions.

1. INTRODUCTION

Techniques for image retrieval follow two main approaches [8]: Text-Based Image Retrieval (TBIR) and Content-Based Image Retrieval (CBIR). TBIR techniques uses descriptions provided by textual annotation, which may introduce inconsistencies due to the human annotator. This is due to the fact that, in many domains, text cannot accurately capture the visual attributes of an image based on human perception. CBIR techniques, in turn, use content-based description instead of textual description. In CBIR, the images are indexed/retrieved considering their extracted visual content, such as color, texture and shape features [1]. Such features together define *feature vectors* containing m elements that are interpreted as points in an m-dimensional space. In the features space, one assumes that a query point is surrounded by points that represent the most similar images to a given image of interest, an operation well-known as similarity query. Such query are appropriately calculated with the application of dissimilarity functions, one of the basis of CBIR.

Despite dealing with inherent information obtained from the images, CBIR systems often present inaccurate results due to the challenging problem of associating low-level features with the high-level semantics of the images. This lack of correspondence between the high-level similarity from the point of view of the user and the low-level image features is known as *semantic gap* [17]. This problem can be caused, for example, by assuming that all features are equally relevant no matter the objective of the image retrieval. In this sense, some features can be very representative for some queries while being irrelevant for other queries. Also, in a given CBIR context, some features have poor or no semantic meaning, while other features are successful in capturing the semantics.

As an attempt to attenuate the semantic gap problem, Relevance Feedback (RF) methods have been proposed [3] [6] [20]. RF methods are very suited to the task of providing to a CBIR system a mechanism that allows it to learn which features best capture the user's interests.

In the RF process, the user is supposed to evaluate the images retrieved in the current query by assigning them values that state their relevance, semantically speaking. After this step the system reformulates the preceding query, taking into account the user's evaluations to improve its results. In many cases the relevant feedback problem itself is handled as a search problem related to weights, parameters, and/or data aggregation models, such as functions combining multiple descriptors. A review on RF for image retrieval is presented in [23].

In regarding search problems, Genetic Algorithms (GAs) provide a general adaptive search methodology based on natural selection; a methodology that has been successfully employed to perform feature selection and weighting on dissimilarity functions used in CBIR systems. In the realm of

[1] This work is based on an earlier work: SAC'12 Proceedings of the 2012 ACM Symposium on Applied Computing, Copyright 2012 ACM 978-1-4503-0857-1/12/03. http://doi.acm.org/10.1145/2245276.2245471.

CBIR systems tuned by Relevance Feedback techniques, this study proposes two GA-based RF methods to enhance the accuracy of image retrieval tasks:

- the first method adjusts the dissimilarity calculus by means of weighting functions that calibrate the importance and impact of each feature in a given features space;
- the second method transforms the features space through linear and non linear transformation functions.

The remainder of this paper is structured as follows. Section 2 presents the related work. Section 3 presents the preliminaries and notations that we use throughout the paper. Section 4 formally describes the proposed methods. Section 5 details the experimental evaluation. Finally, Section 6 presents the conclusions and points out future works.

2. RELATED WORK

Researches on improving image retrieval effectiveness mainly employ RF [23], dissimilarity function learning [21] and feature selection [15] methods. The most used RF approach employs dynamic weighting mechanisms to modify the distance function or image dissimilarity model through appropriate weights, so that the distance between relevant images becomes smaller if compared to the non-relevant images.

In the study of Stejic *et al.* [18], the authors incorporate GA into RF mechanisms in order to assign the appropriate weights on image descriptors and image regions. However, the authors did not provide an effective model to learn the user's requests, because the *R-precision* evaluation function that they employed represents only the ratio of retrieved relevant images. Differently, our study addresses this question through a GA-based RF mechanism that relies on an order-based ranking evaluation function.

Based on the premise that dissimilarity functions and image descriptors are problem-oriented, Torres *et al.* [21] proposed the use of nonlinear combinations of multiple image similarities, addressing the problem through the use of Genetic Programming (GP). In their investigation, the learning process relies on a training set and not on the user's feedback. Thus, the outcomes of their methods are not adjusted to the user's transient interests.

Other studies attempt to improve the precision of CBIR systems by working directly with the features space [13] [5]. The study of Silva *et al.* [5] relies on ranking evaluation functions in order to choose the best set of features to represent images in the CBIR context; the contribution of each feature is binary (selected or not selected). In a different course of action, our methods take advantage of the relative importance of each feature in image retrieval, considerably improving the retrieval results.

3. PRELIMINARIES AND NOTATION

Let $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ represent the set of feature vectors extracted from the image database $\mathbf{I} = \{\mathbf{i}_1, \ldots, \mathbf{i}_n\}$ using the feature extractor ε, i.e., $\varepsilon(\mathbf{i}_i) = \mathbf{x}_i = \{x_1, \ldots, x_m\}$, $\mathbf{x}_i \in \mathbb{R}^m$. Now consider $\mathbf{x}_q$ as being the feature vector extracted from the query image $\mathbf{i}_q$ and $\mathbf{x}_i$ the feature vector extracted from an arbitrary image $\mathbf{i}_i$. A dissimilarity function $d()$, also called distance function, provides a mechanism to measure the dissimilarity between $\mathbf{x}_q$ and $\mathbf{x}_i$.

Since the smaller the dissimilarity the larger the similarity is, the elements of $\mathbf{X}$ can be sorted according to their similarities to $\mathbf{x}_q$. In CBIR the most popular form of similarity query is the k-Nearest Neighbor query, or kNN query for short.

The kNN query aims at finding the k closest objects to a query object [22]. A kNN query is formally presented in Definition 1.

Definition 1. Given the set of objects $\mathbf{X}$, a distance function $d()$ and the query object $\mathbf{x}_q$, the response set $\mathbf{T}$ of a kNN query is defined as:

$$kNN(\mathbf{x}_q, k, d(), \mathbf{X}) = \{\mathbf{T} \subseteq \mathbf{X}, |\mathbf{T}| = k \wedge \forall \mathbf{x}_i \in \mathbf{T}, \mathbf{x}_j \in \mathbf{X} \setminus \mathbf{T} : d(\mathbf{x}_q, \mathbf{x}_i) \leq d(\mathbf{x}_q, \mathbf{x}_j)\} \quad (1)$$

We consider that the elements of $\mathbf{T}$ are sorted according to their distances to the query image composing a ranking $\mathcal{T} = (\mathbf{t}_1 \prec \mathbf{t}_2 \prec \ldots \prec \mathbf{t}_{k-1} \prec \mathbf{t}_k)$, where $\forall i = \{2, \ldots, k\}$, $d(\mathbf{x}_q, \mathbf{t}_{i-1}) \leq d(\mathbf{x}_q, \mathbf{t}_i)$.

In order to evaluate the effectiveness of kNN queries into the optimization process of RF, we employed a ranking quality measure, as described in Definition 2. This is an order-based utility function that considers the utility of images retrieved according to their ranks [11]. Relevant images in the first positions of the ranking will receive higher scores of utility while relevant images far from the ranking top will receive lower scores.

Definition 2. Given the ranking $\mathcal{T}_i = (\mathbf{t}_1, \ldots, \mathbf{t}_k)$ as the result of the query $kNN_i(\mathbf{x}_q, k, d(), \mathbf{X})$ and $\mathbf{R}_q = \{\mathbf{r}_1, \ldots, \mathbf{r}_\rho\}$, $\mathbf{R}_q \subseteq \mathbf{X}$ the set of objects that belong to the same class of $\mathbf{x}_q$, also called here the relevant objects for $\mathbf{x}_q$, the measure of the quality of $\mathcal{T}_i$ is calculated by the function:

$$\Phi(\mathcal{T}_i, \mathbf{R}_q) = \sum_{j=1}^{k} \frac{r(\mathbf{t}_j)}{A} . \left(\frac{(A-1)}{A}\right)^{j-1} \quad (2)$$

where $r(\mathbf{t}_j) = 1$, if $\mathbf{t}_j \in \mathbf{R}_q$ or $r(\mathbf{t}_j) = 0$ otherwise, and $A \geq 2$ is an adjustment parameter that expresses the relative importance of the position of the elements on the ranking. Small values for A means more importance for the relevant elements on the first positions of the ranking. When A is large, the fraction $\frac{(A-1)}{A}$ is close to 1, then the position of the elements on the ranking is not strongly considered.

We apply the proposed ranking-quality measure as the fitness function at the core of the Genetic Algorithm used along this research. Our goal relies on achieving a ranking that maximizes the value of this function.

4. PROPOSED METHODS

4.1 Overview of the System

The scheme of the system is shown in Figure 1. We suppose that we have an image data set $\mathbf{I}$ with the image features extracted using a feature extractor ε. Initially, the user enters a query image $\mathbf{i}_q$ and we apply ε on $\mathbf{i}_q$ to get the feature vector $\mathbf{x}_q$, i.e. $\mathbf{x}_q = \varepsilon(\mathbf{i}_q)$. Then, the system computes the distance between the respective feature vector $\mathbf{x}_q$ to the features of the images from data set $\mathbf{I}$.

After this, a ranking is generated and presented to the user, who is in charge of evaluating the first k images in the ranking, assigning them relevance values: *relevant*, $r(\mathbf{t}_j) = 1$ if the image is relevant to the query; and *not desirable*, $r(\mathbf{t}_j) = -1$ if the image is not supposed to be in the ranking. By default, the system assigns *not relevant*, $r(\mathbf{t}_j) = 0$ to every image that eventually has not been evaluated by the user. This iterative process defines the set of relevant objects $\mathbf{R}_q$. Since many images can belong to multiple classes even according to the user's judgment, it is the specific user's need that will define whenever an image is related or not to the image query $\mathbf{i}_q$.

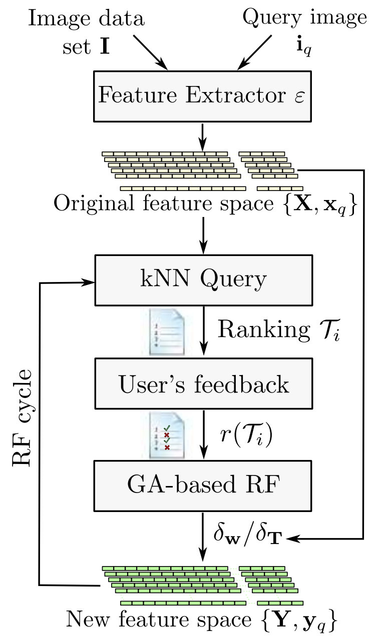


Image data set $\mathbf{I}$
Query image $\mathbf{i}_q$
Feature Extractor ε
Original feature space $\{\mathbf{X}, \mathbf{x}_q\}$
kNN Query
Ranking $\mathcal{T}_i$
User's feedback
$r(\mathcal{T}_i)$
GA-based RF
$\delta_{\mathbf{w}}/\delta_{\mathbf{T}}$
RF cycle
New feature space $\{\mathbf{Y}, \mathbf{y}_q\}$

Figure 1: Systematization of the proposed method.

Once the user's feedback is provided, we apply a GA-based search for a sequence of *weighting functions* or *transformation functions* ($\delta_{\mathbf{w}}$ or $\delta_{\mathbf{T}}$) that that will maximize the fitness function presented in Equation 2. The application of δ_w and δ_T to adjust the content-based retrieval are presented formally in subsections 4.3 and 4.4, respectively. In summary, a sequence of weighting functions is inferred in order to adjust the distance function so to reflect the user feedback; and a sequence of transformation functions is determined in order to transform the original feature space $\mathbf{X}$ and the feature vector of the query $\mathbf{x}_q$ so to achieve more accuracy in retrieval tasks.

Values of relevance provided by the user are stored in a temporary space in between successive RF iterations. The cycles of RF/GA-search are repeated until the optimal solution is found, which means that the resulting ranking contains, as much as possible, only relevant images in its first k positions; or until a predefined maximum number of cycles is reached.

In short, the approaches we propose are:

- inferring a weight vector by means of weighting functions; this allows the distance function to take into account the degree of contribution of each feature according to the user's feedback;
- optimizing the features space by inferring a space transformation that will adjust the original space in order to better represent the user's similarity criteria.

4.2 Weighting Functions Approach

The use of GA to infer the most appropriate weight vectors for the dissimilarity function has achieved promising results as pointed out in previous studies. Accordingly, our first approach is based on this same concept but, in an innovative fashion, we generate weights that obey to well-known mathematical functions, as opposed to former methods that use constant values. This proposal came from investigating the distribution of the dissimilarity distances before and after the use of a weight vector. The assumption was that there could be a well-defined manner to determine the best weight to each feature so to improve the ranking generated by the similarity queries. In accordance, we implemented a CBIR system that integrates Relevance Feedback and Genetic Algorithms in order to find weight vectors in the form of sequences of mathematical functions. For this intent, we considered a vast set of functions, as illustrated in Figure 2.

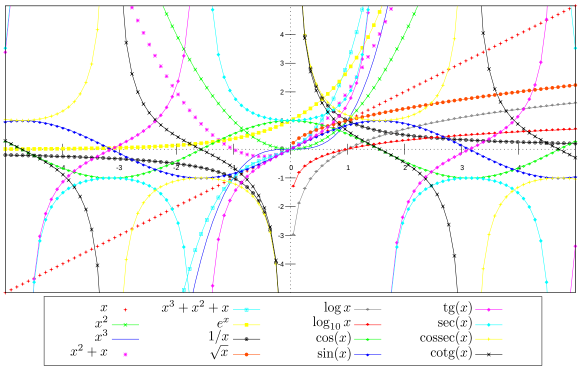


Figure 2: Examples of non linear and linear functions used to generate the inference of weights and features transformations.

The problem, formally presented in Definition 3, consists in finding a sequence of functions that generate the best weight vector for a specific query. In our problem, the chromosome used by the GA search is encoded slightly different from the most usual approach, according to which real values corresponding to the weights are assigned to genotype. In our case, the values assigned to the genotype are integers that correspond to the identifiers of the weighting functions.

Definition 3. Let $\mathbf{F} = \{f_1, \ldots, f_\rho\}$ be a set of mathematical functions and $\delta_\mathbf{w} = (f'_1, \ldots, f'_m)$, $f'_i \in \mathbf{F}$ be the ordered set of weighting functions inferred. The number of elements of the sequence δ_w must be equal to the number of features employed for images descriptions. Now let $\mathbf{w} = \delta_\mathbf{w}(\mathbf{x}_q) = \{f'_1(x_{q_1}), \ldots, f'_m(x_{q_m})\}$ be the weighting vector for the distance function $d_\mathbf{w}()$, where $f'_i(\mathbf{x}_{qi})$ is a transformation applied on the i-th feature of query image $(\mathbf{x}_{qi})$ to produce a weight for the i^{th} feature in the dissimilarity function. The problem using the weighting functions is to find a set $\delta_\mathbf{w}$ such that

$$\arg\max_{\delta_\mathbf{w}} \Phi(\mathcal{T}, \mathbf{R}_q) \tag{3}$$

where $\mathcal{T} = kNN(\mathbf{x}_q, k, d_\mathbf{w}(), \mathbf{X})$ and $\mathbf{R}_q$ is the relevance values for each image regarding the query represented by feature vector x_q.

The advantage of dealing with weighting functions is that the search space for the GA is significantly reduced in comparison with the usual weighting over a given interval, such as $[0, 1]$. The search space in a continuous interval is much higher than the discrete and finite search space $\mathbf{F}$, according to, for each weight, we have only $|\mathbf{F}|$ possible choices.

4.3 Feature Space Transformation Functions Approach

The second approach we introduce is based on the transformation of the features space, according to which each feature value is redefined by a transformation function, as detailed in Definition 4. These functions provide linear and nonlinear transformations over the original features space, a way to capture nonlinear relationships.

Transformations over the features space are defined using the GA search to find the sequence of functions that lead to improved retrieval results. For each sequence of transformation functions considered by the GA, a new feature space is calculated using the original feature values; after each consideration, a new kNN query is executed.

Definition 4. Given $\mathbf{F}$, as previously set out, let $\delta_\mathbf{T}$, defined analogously to $\delta_\mathbf{w}$, be an ordered set of inferred transformation functions and $\mathbf{Y} = \{\mathbf{y}_1, \ldots, \mathbf{y}_n\}$ the new feature space, and each $\mathbf{y}_i$ corresponding to a $\mathbf{x}_i$ transformed by the functions in $\delta_\mathbf{T}$, i.e., $\mathbf{y}_i = \delta_\mathbf{T}(\mathbf{x}_i) = \{f'_1(x_{i_1}), \ldots, f'_m(x_{i_m})\}$.

The problem translates into finding the set $\delta_\mathbf{T}$ of transformation functions such that:

$$\arg\max_{\delta_\mathbf{T}} \Phi(\mathcal{T}, \mathbf{R}_q) \tag{4}$$

where $\mathcal{T} = kNN(\mathbf{y}_q, k, d(), \mathbf{Y})$.

4.4 Genetic Algorithm Description

When implementing a GA, it is necessary to consider its parameters and operators, such as the chromosome coding, the fitness function, selection, the crossover, and the mutation operators. The values of the parameters and the chosen operators can be decisive regarding the effectiveness of the algorithm. In this investigation, such choices were made experimentally and are described as follows:

- *Chromosome coding*: for the two proposed methods, a chromosome was coded as an integer-valued vector with m positions, $\mathbf{C} = (g_1, \ldots, g_m)$, where g_i corresponds to an identifier of a function in $\mathbf{F}$. In the weighting functions approach, the chromosome produces the weight vector $\mathbf{w}$ for the distance function $d_\mathbf{w}()$; while in the features space transformation approach, it provides the new features space $\mathbf{Y}$.

- *Fitness function*: as fitness function we employed the ranking quality measure presented in Definition 2.

- *Selection for recombination operator*: we used exponential ranking selection to select pairs of individuals to reproduce. For an individual $\mathbf{C}_i$, the probability p_i of being selected is given by Equation 5:

$$p_i = \frac{c^{Sp-i}}{\sum_{j=1}^{Sp} c^{Sp-j}}, 0 \leq c \leq 1 \tag{5}$$

  where $i \in \{1, \ldots, Sp\}$, Sp is the population size and $c = 0.9$.

- *Selection for reinsertion operator*: elitism was employed to select the surviving chromosome for the next generation. This is because elitism guarantees that the best individual of the population in a generation g will be present in the population of generation $g+1$, and so it guarantees that the best solution will be improved or, at least, maintained.

- *Crossover operator*: we employed uniform crossover. A mask is randomly built and it indicates which chromosome will supply each gene for the first offspring. The second offspring is generated by the complement of the same mask.

- *Mutation operator*: uniform mutation is applied on the offspring chromosomes. The genes are selected with probability P_m and their values are changed by another valid value randomly chosen.

5. EXPERIMENTAL RESULTS AND ANALYSIS

We evaluated our proposed methods by comparing it with the weighting approach most commonly used, according to which the weights correspond to values in the range $[0, 1]$ – here, we named this approach *direct weights generator (WG)*. This section describes the setting of the experiments and the analysis of the results.

5.1 Image Data sets

In order to assess the applicability of the proposed methods, experiments were conducted on real data sets in two different domains: medical and general domain.

General Domain

The first image database from a general domain is *Corel 1000* [9]. This data set contains 1000 images classified into 10 classes with 100 images per class (*africa*, *beach*, *buildings*, *buses*, *dinosaurs*, *elephants*, *flowers*, *food*, *horses* and *mountains*).

The other general domain data set used for experiments is *Scenes* [14]. This data set is comprised of 1678 images divided into 5 classes: 2 classes of urban environments (*highway* with 260 images and *tall buildings* with 356 images), and 3 classes of natural environments (*coast* with 360 images, *forest* with 328 images, and *mountain* with 374 images).

Medical Domain

From the medical domain we used two data sets, one related to general exams and one more specific. These data sets were collected at the Clinical Hospital at University of São Paulo in Ribeirão Preto.

The first data set called *Medical Exams* is a collection of 2400 medical images of various body parts. The images were extracted by X-Ray and MRI exams, classified according to the body part and cut type. The base is split into 12 classes (*abdomen*, *axial brain*, *coronal brain*, *sagittal brain*, *breast*, *chest*, *leg*, *hand*, *knee*, *lung*, *pelvis* and *spine sagittal*), each class contain 200 images.

The second data set, called *Lung*, consists of 246 MRI exams of human lungs. This data set is divided into 6 classes being 1 of normal and 5 of abnormal patterns (*consolidation*, *emphysema*, *thickness*, *honeycombing*, and *ground-glass opacity*), with an average of 41 images per class, varying from 39 to 44.

5.2 Parameters Setup

Feature Extractors

For each data set, the feature vectors were acquired by the feature extractors: *Color Moments* [19], *Co-occurrence* [7], *Sobel Histogram* [2], *Histogram* [4], *Run Length* [10] and *SIFT* [12]. Table 1 shows the number of features extracted and the type of the information captured by each extractor. When extracting *Color Moments* and *Sobel Histogram*, the images were partitioned into 16 rectangular regions. The respective features were extracted from each region and combined in a single vector.

Table 1: Feature extractors employed

Feature extractor	Number of features	Type
Color Moments	144	Color
Co-occurrence	88	Texture
Sobel Histogram	128	Shape
Histogram	256	Color
Run Length	44	Texture
SIFT	128	Shape

All the features were normalized using the *z-score* function to avoid *bias* on distance calculation. The dissimilarity measures were obtained by the Euclidean distance function (L_2). The Weighted Euclidean distance function was applied to the weighting methods.

One important issue concerned to the feature extractors is that if the extracted features do not describe the relevant aspect of the user's interests, the GA may not converge to satisfactory solutions. This is due to the fact that the set $\mathbf{R}_q$ will be empty and therefore won't be able to contribute to the fitness computation.

GA Parameters

After some previous experiments and analysis, it was observed that the values assigned to the GA parameters that achieved better results were:

- population size (Sp): 50;
- maximum number of generations (Ng): 100;
- crossover rate (Pc): 0.8;
- mutation rate (Pm): 0.02.

5.3 Results

All the experiments were performed using the three following methods: direct weights generator (WG), weighting functions (WF) and transformation functions (TF). All the methods (WG, WF e TF) employ the fitness function of Equation 2, over which we test the values 2, 10 and 20 for parameter A. Also, we analyze the performance of the different feature extractors. Regarding the kNN query, we empirically selected $k = 30$ and the maximum number of cycles of RF/GA-search is 10.

The effectiveness of our methods WF and TF, in comparison with the WG method, was assessed by method Precision *vs.* Recall (P&R), by method Number of Relevant Images *vs.* Cycles of RF, and by visual data analysis. The qualitative visual analysis aimed at verifying the distribution of the relevant and non relevant images in the resulting optimized spaces, and aimed at measuring the cohesion of the cluster composed of relevant images.

The visual data analysis employed here was obtained with tool *Metric Space Platform* (*MetricSplat*) [16]. This tool combines the benefits of visualization techniques with methodologies for content-based image retrieval. Fastmap projection (cloud of point) was the technique chosen to illustrate features space configuration before and after optimizations.

In the visualization process, for each data set one image was randomly chosen. Then, we compared the results of the initial query and the results given after applying the optimization methods WF and TF. In order to allow a better visual analysis, for each visualization, we considered the entire data space and, comparatively, a reduced space with only the first 50 elements. Red dots represent the relevant elements of the query and the blue dots represent elements that are not relevant.

The cohesion measure was used to quantify how close are the images that belong to the same class of the query in relation to the query center. The cohesion measures were taken before and after the optimizations. We used measure Mean Square Deviation (MSD), calculated as follows:

$$MSD(C) = \frac{1}{n}\sum_{i=1}^{k}(c - x_i)^2 \tag{6}$$

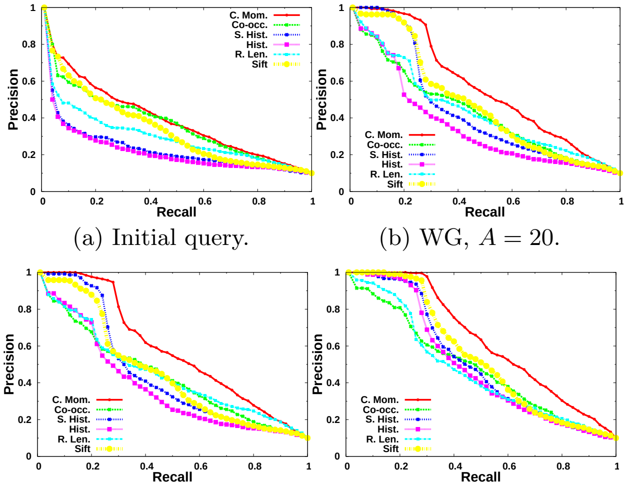
where c is the centroid (the query in this case) and x_i is an element of cluster C (here it is considering only the cluster composed of relevant elements). Small values for MSD indicate better cohesion on clusters.

5.3.1 *Corel 1000*

Experiments on *Corel 1000* are presented considering an average of 10 image queries randomly chosen, one from each class. Figure 3(a) shows low precision of each feature extractor in the initial query and the improvement on precision obtained in the last RF cycle using each method (Figures 3(b), (c) and (d)). It can be observed that the WF (Figure 3(c)) obtains higher precision then WG (Figure 3(b)), near to 20% of recall with *Color Moments* feature extractor; TF (Figure 3(d)), in turn, achieved the highest precision, near to 30% of recall, also using *Color Moments*.

Considering extractor *Color Moments*, which was the best extractor for the *Corel 1000* data set, in Figure 4(a) we analyze the adjustment of the parameter A of the fitness function for each method. Figure 4(a) shows that the best results were achieved when using TF, with $A = 20$, $A = 10$ and $A = 2$, respectively. Following, one can see the WF method with $A = 20$ and $A = 10$. In Figure 4(b) it is possible to observe that the precision increases until the fifth cycle, with low or no improvement after that. We believe that the Color Moments were the best extractor for the Corel 1000 data set because it comprises image classes each with a characteristic color hue.

The number of relevant images retrieved per cycle obtained by each feature extractor using TF method with $A = 20$ is shown on Figure 5(a). It can be observed that the texture-based feature extractors, *Co-occurrence* and *Run Length*, have similar and poor improvement while compared to the other ones. Figure 5(b) shows the number of relevant images retrieved through the cycles for each method with $A = 2, 10, 20$ and using the *Color Moment* extractor. As we can see, the TF method was the most effective, followed by the WF method for the average of all feature extractors.



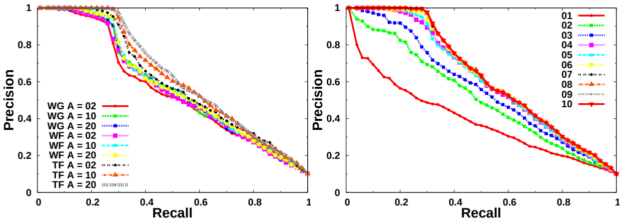
(a) Initial query.

(b) WG, $A = 20$.

(c) WF, $A = 20$.

(d) TF, $A = 20$.

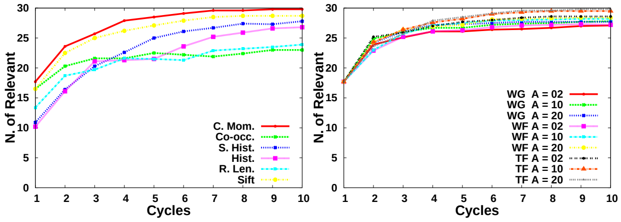
Figure 3: P&R plot for each feature extractor and $A = 20$ (a) initial query, (b) cycle 10 using WG, (c) cycle 10 using WF, (d) cycle 10 using TF.



(a) Cycle 10.

(b) TF, $A = 20$.

Figure 4: P&R plot for the feature extractor *Color Moments* (a) cycle 10 for each method and for each value of A, (b) evolution through the cycles 01 to 10 using TF and $A = 20$.



(a) TF, $A = 20$.

(b) *Color Moments*.

Figure 5: Number of Relevant (a) for each feature extractor using TF and $A = 20$, (b) using *Color Moments* for each method and for each value of A.

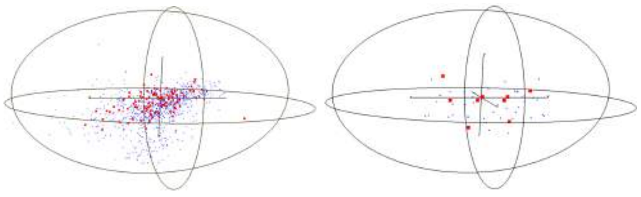
Figure 6 shows the projection of the original space from *Corel 1000* using *Color Moments*. The configuration of the spaces after using methods WG and WF are respectively illustrated in Figure 7 and Figure 8. Figures 7(a) and 8(a) show that the generated spaces preserve a sparse distribution of the elements, similar to the original space (Figure 6(a)). WG retrieved 16 relevant elements (Figure 7(b)) while WF retrieved 17 (Figure 7(b)) among the first 50 elements retrieved. Notice that, in the visualizations, the space scales are not fixed.

Comparing the TF method with the weighting approaches, it can be seen (Figure 9(a)) that the generated space is more compact, and the relevant elements are concentrated closer to the query center. Figure 9(b) shows that TF retrieved 33 relevant images of the 50 first elements in the ranking; 94% more accurate than WG and WF.

The bar chart in Figure 10 shows the cohesion for each space configuration. The cluster of relevant images obtained by the initial query had the higher cohesion; meanwhile, the cluster obtained by TF presented the best value. Considering the weighting methods, WF was superior than WG on its space configuration.

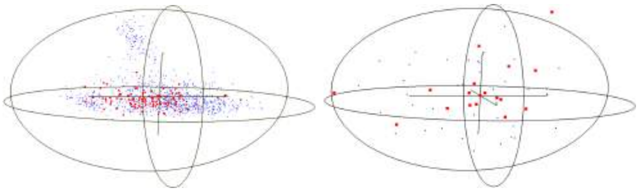
5.3.2 *Scenes*

Figure 11(a) illustrates the precision *vs.* recall results for the initial queries on data set *Scenes*. It can be seen that the *Co-occurrence* extractor achieved the best result while *Histogram* achieved the worst result. After 10 optimization cycles, it can be observed in Figures 11(b), (c) and (d) that the extractor *Sobel Histogram* achieved the best results for all methods. WG was more accurate when assigned $A = 20$, while WF and TF had better results with $A = 10$. Fixing recall at 20%, the average precision obtained by TF was 67%,



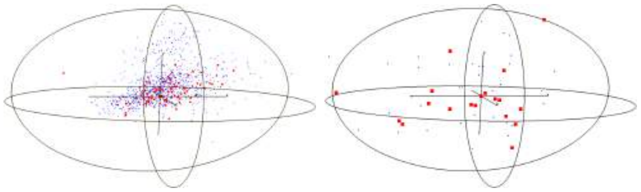
(a) (b)

Figure 6: Original space configuration of *Corel 1000* using *Color Moments* (a) entire space (b) 50 nearest elements from the query point.



(a) (b)

Figure 7: Data space configuration of *Corel 1000* data set after applying WG method (a) entire space (b) 50 nearest elements from the query point.



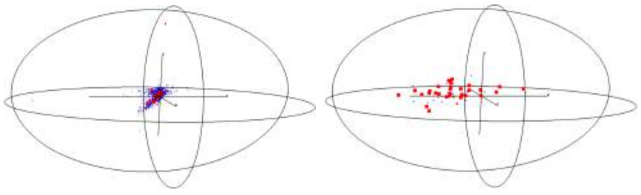
(a) (b)

Figure 8: Data space configuration of *Corel 1000* data set after applying WF method (a) entire space (b) 50 nearest elements from the query point.

superior to WF with 64% and WG with 61%. The Sobel Histogram achieved the best results because this descriptor can capture the complex patterns present in images with a high number of edges, which is the case for the Scenes data set.

Regarding extractor *Sobel Histogram*, Figure 12(a) shows the results of the last cycle for each method and the values assigned to parameter A. The maximum precision was achieved by methods TF ($A = 2$, 10 and 20) and WF ($A = 10$) up to the recall rate of 8%. As Figure 12(b) shows, the improvements for method TF with $A = 10$ were more significant until the fourth cycle, that is, the method converged at this cycle.

The graphics in Figure 13 present the number of relevant images retrieved per cycle. Figure 13(a) shows results obtained by TF with $A = 10$, where the more effective extractor was the *Sobel Histogram*, followed by *SIFT* on the last cycles. For extractor *Sobel Histogram*, the methods TF and WF (as illustrated on Figure 13(b)) were slightly more effective while using $A = 10$, in contrast to WG, which was more effective with $A = 20$. TF achieved an average of 30 relevant retrieved images, while WF and WG achieved 29 and 28, respectively.



(a) (b)

Figure 9: Data space configuration of *Corel 1000* data set after applying TF method (a) entire space (b) 50 nearest elements from the query point.

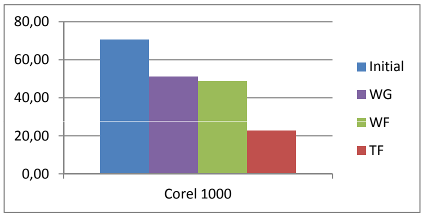
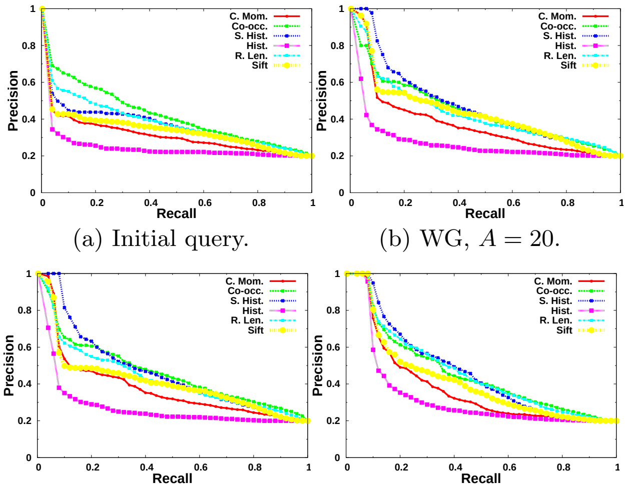


Figure 10: MSD values for relevant elements clusters on *Corel 1000* data set. The smaller the MSD value the higher cohesion.



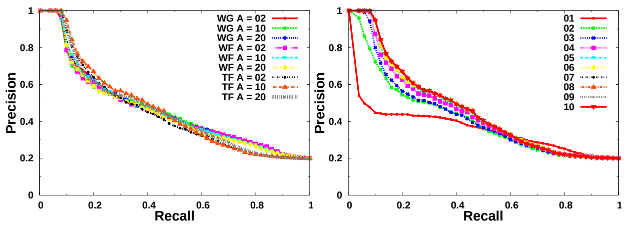
(a) Initial query. (b) WG, $A = 20$.

(c) WF, $A = 10$. (d) TF, $A = 10$.

Figure 11: P&R plot for each feature extractor (a) initial query, (b) cycle 10 using WG, $A = 20$, (c) cycle 10 using WF, $A = 10$, (d) cycle 10 using TF, $A = 10$.

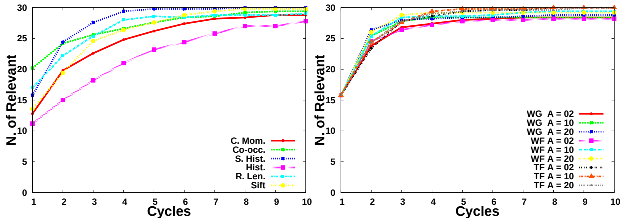
Figure 14 shows the visualization of the original data space using *Sobel Histogram*. The space obtained by TF (Figure 17(a)) was better adjusted to the query because the relevant elements were closer to the center, different from the spaces obtained by WG and WF (Figures 15(a) and 16(a)).

All optimization methods increased the number of relevant images considering the 50 elements closer to the center, as Figures 15(b), 16(b) and 17(b) show. The best results were obtained by WF and TF, both with 36 relevant elements in the first 50 elements, while WG achieved 21 relevant elements. Figure 18 shows the MSD values for the cluster with relevant elements. TF has a slightly better result if compared to WF and to WG; all the three methods had a significantly smaller MSD for the relevant cluster, which means that cohesion increased and so the optimizations were very effective.



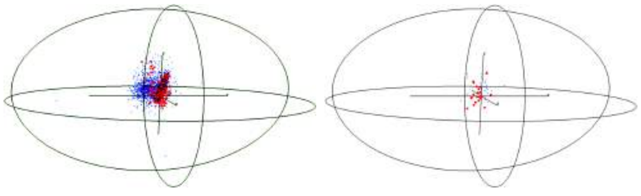
(a) Cycle 10. (b) TF, $A = 10$.

Figure 12: P&R plot for the feature extractor *Sobel Histogram* (a) cycle 10 for each method and for each value of A, (b) evolution through the cycles 01 to 10 using TF and $A = 10$.



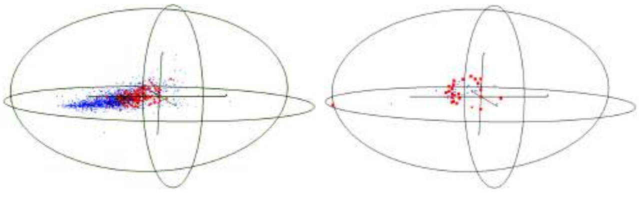
(a) TF, $A = 10$. (b) *Sobel Histogram.*

Figure 13: Number of Relevant (a) for each feature extractor using TF and $A = 10$, (b) using *Sobel Histogram* for each method and for each value of A.



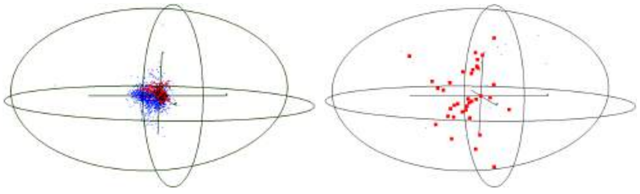
(a) (b)

Figure 14: Original space configuration of *Scenes* using *Sobel Histogram* (a) entire space (b) 50 nearest elements from the query point.



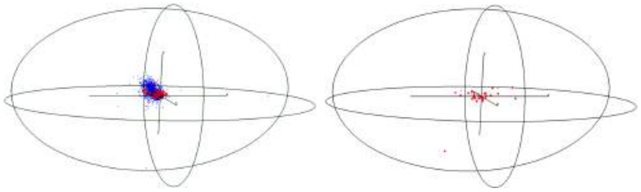
(a) (b)

Figure 15: Data space configuration of *Scenes* data set after applying WG method (a) entire space (b) 50 nearest elements from the query point.



(a) (b)

Figure 16: Data space configuration of *Scenes* data set after applying WF method (a) entire space (b) 50 nearest elements from the query point.



(a) (b)

Figure 17: Data space configuration of *Scenes* data set after applying TF method (a) entire space (b) 50 nearest elements from the query point.

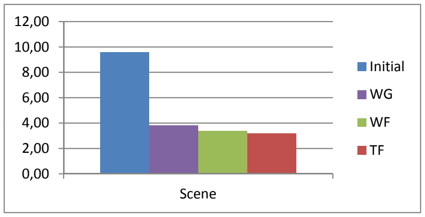


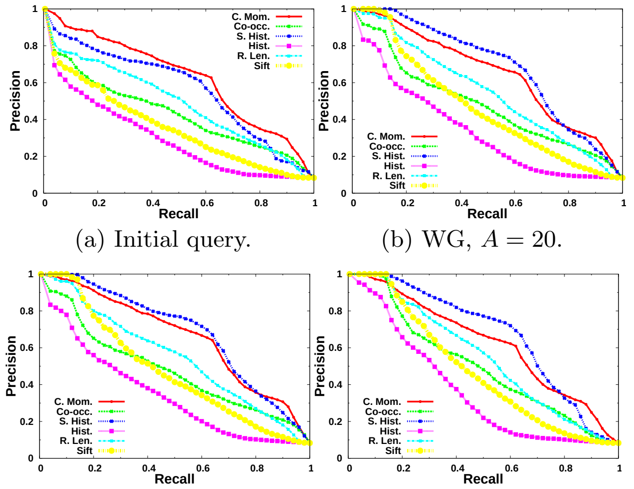
Figure 18: MSD values for relevant elements clusters on *Scenes* data set.

5.3.3 Medical Images

The average P&R results for the initial queries on the *Medical Images* data set are shown in Figure 19(a). *Color Moments* presented the best initial results, with an average of 26 relevant images retrieved, whilst *Histogram* obtained the worst initial results, as it retrieved an average of 17 relevant images considering the first 30 images retrieved. For this data set, extractor *Sobel Histogram* was more accurate after the optimization using WG ($A = 20$), WF ($A = 10$) and TF ($A = 10$) methods, as can be seen in Figures 19(b), (c) and (d), respectively). Considering recall rate at 20%, the average precision was near 95% for all methods. Texture-based extractors achieved the best results after optimization using TF; this result is expected as medical images are highly characterized by textural content.

All three methods achieved maximum precision while using *Sobel Histogram* up to 14% of recall, as it is shown in Figure 20(a). Improvements on accuracy through the cycles while using TF and $A = 10$ are shown in Figure 20(b). Once more, the convergence occurs around the fourth cycle.

The plots of the number of relevant results per cycle in Figure 21 show that extractor *Sobel Histogram* had the best response regarding TF optimization, followed by *SIFT*(Figure 21(a)). Figure 21(b) also shows that all meth-



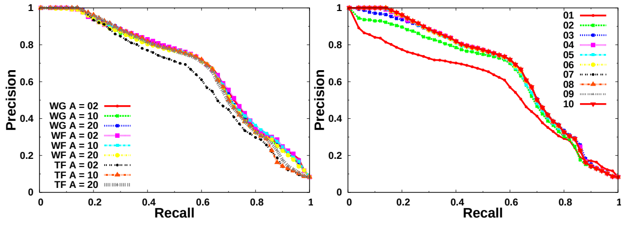
(a) Initial query.

(b) WG, $A = 20$.

(c) WF, $A = 10$.

(d) TF, $A = 10$.

Figure 19: P&R plot for each feature extractor and (a) initial query, (b) cycle 10 using WG, $A = 20$, (c) cycle 10 using WF, $A = 10$, (d) cycle 10 using TF, $A = 10$.

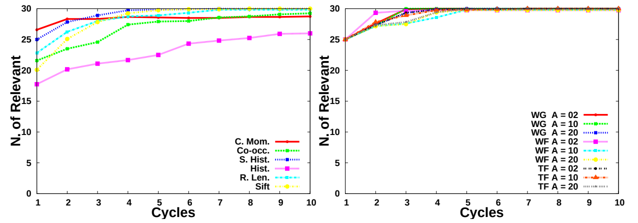


(a) Cycle 10.

(b) TF, $A = 10$.

Figure 20: P&R plot for the feature extractor *Sobel Histogram* (a) cycle 10 for each method and for each value of A, (b) evolution through the cycles 01 to 10 using TF and $A = 10$.

ods achieved 30 relevant images considering the 30 first images retrieved in the fifth cycle while optimizing features extracted by *Sobel Histogram*.



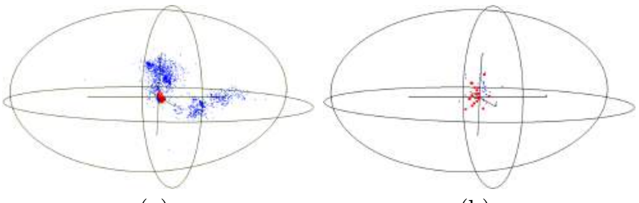
(a) TF, $A = 10$.

(b) *Sobel Histogram.*

Figure 21: Number of Relevant (a) for each feature extractor using TF and $A = 10$, (b) using *Sobel Histogram* for each method and for each value of A.

The visualizations of the *Medical Images* data set correspond to the spaces generated by the features extracted using *Sobel Histogram*. Figure 22 illustrates the initial configuration of this space. It is noticeable that there is a concentration of relevant images near to the center of the space (Figure 22(a)).

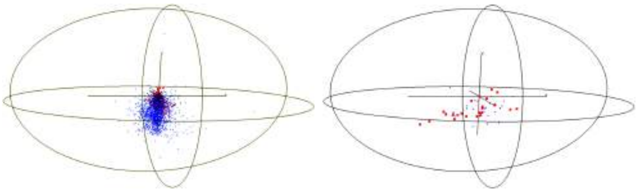
The distribution of the elements on space generated by WG (Figure 23(a)) reveals that the method brought the non relevant images closer to the center in comparison to the initial space. The methods WF (Figure 24(a)) and TF (Figure 25(b)) generated better configurations. Both methods preserved a separation of non relevant elements. TF was more effective since the cluster of the relevant images was very close to the center. Considering the 50 elements closer to the center, WF retrieved 34 relevant ones (Figure 24(b)) and TF (Figure 25(b)) retrieved 36, while WG retrieved 25 (Figure 23(b)). Measures of cohesion on the relevant clusters confirm the visualizations. As shown in Figure 26, TF obtained the lowest value, with significant difference from the values on WG and WF.



(a)

(b)

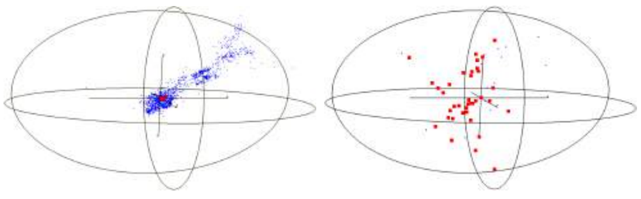
Figure 22: Original space configuration of *Medical Images* using *Sobel Histogram* (a) entire space (b) 50 nearest elements from the query point.



(a)

(b)

Figure 23: Data space configuration of *Medical Images* data set after applying WG method (a) entire space (b) 50 nearest elements from the query point.



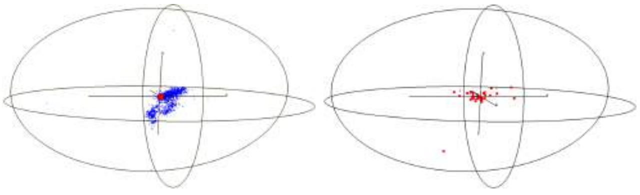
(a)

(b)

Figure 24: Data space configuration of *Medical Images* data set after applying WF method (a) entire space (b) 50 nearest elements from the query point.

5.3.4 Lung

For this data set we have randomly chosen 6 images, one from each class; following we present and the average results obtained with the respective queries. Figure 27(a) shows the low precision of each feature extractor of the initial query, with the best performance, again, obtained by using *Color Moments*. Figure 27(b) and 27(c) illustrate the



(a) (b)

Figure 25: Data space configuration of *Medical Images* data set after applying TF method (a) entire space (b) 50 nearest elements from the query point.

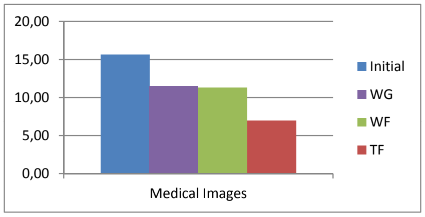
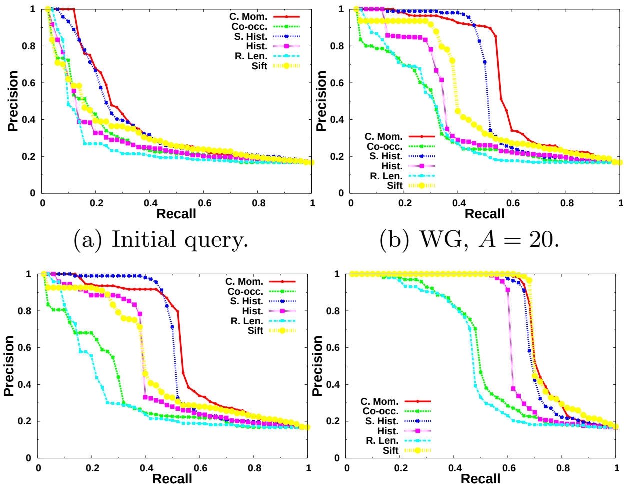


Figure 26: MSD values of clusters of relevant elements on *Medical Images* data set.

performance of WG and WF methods respectively, with a slightly better effectiveness of WF and best improvement achieved by the feature vector *Sobel Histogram* in both cases. Figure 27(d) shows significantly higher effectiveness of TF method (with 100% of precision until 60% of recall) in comparison to both WG and WF, and the best improvement achieved using *SIFT*.

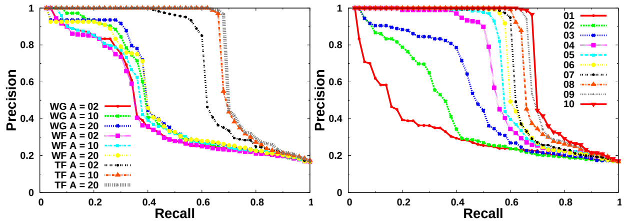


(a) Initial query. (b) WG, $A = 20$.

(c) WF, $A = 20$. (d) TF, $A = 20$.

Figure 27: P&R graphics for each feature extractor and $A = 20$ (a) initial query, (b) cycle 10 using WG, (c) cycle 10 using WF, (d) cycle 10 using TF.

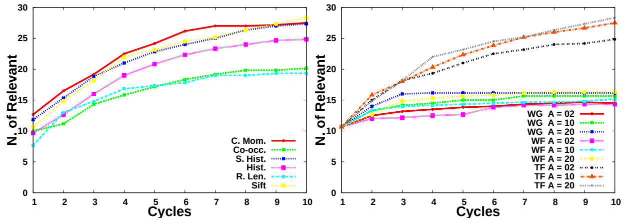
Considering the feature extractor *SIFT*, Figure 28(a) shows the highest precision when using TF and $A = 20$, 10 and 2, in decreasing order. Furthermore, the performance difference in using weighting and transformation techniques is noticeable. For instance, taking 60% of recall, the transformation approach achieved 100% of precision, while weighting was around 30% of precision. The results on each cycle, using *SIFT*, TF and $A = 20$, are illustrated on Figure 28(b). Once more, after the fifth cycle low improvements are achieved.



(a) Cycle 10. (b) TF, $A = 20$.

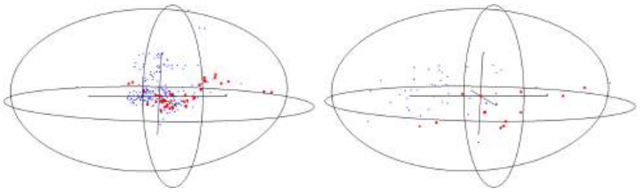
Figure 28: P&R graphics for the feature extractor *SIFT* (a) cycle 10 for each method and for each value of A, (b) evolution through the cycles 01 to 10 using TF and $A = 20$.

Figure 29(a) shows that, also in this data set, the texture-based feature extractors (*Co-occurence* and *Run Length*) achieved the lowest effectiveness on TF method. Considering the *SIFT*, Figure 29(b) illustrates that the TF method significantly outperforms the others, and WF with $A = 20$ has a slightly better result than the WG method in the last three cycles.



(a) TF, $A = 20$. (b) *SIFT*.

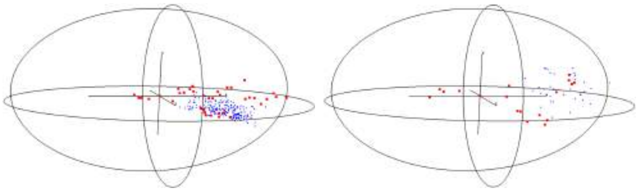
Figure 29: Number of Relevant (a) for each feature extractor using TF and $A = 20$, (b) using *SIFT* for each method and for each value of A.



(a) (b)

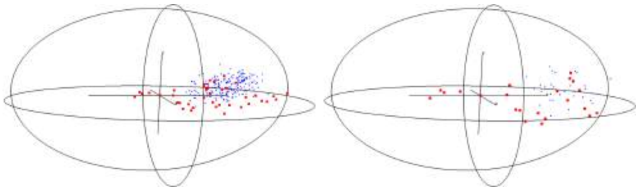
Figure 30: Original space configuration of *Lung* using *SIFT* (a) entire space (b) 50 nearest elements from the query point.

In the visualization and cluster cohesion analysis, *SIFT* was the extractor used in *Lung* data set. Figure 30 shows the distribution of the initial data space. The data spaces obtained from WG, WF and TF methods are illustrated in Figures 31, 32 and 33. In fact, for this data set, all three methods generated a similar distribution, where the relevant elements are more spread than the non relevant elements. Nevertheless, the best configuration after optimization was obtained by TF, since the non relevant images were closer to each other and more distant to the center. SIFT was the best extractor



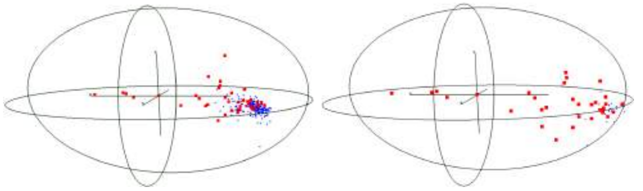
(a) (b)

Figure 31: Data space configuration of *Lung* data set after applying WG method (a) entire space (b) 50 nearest elements from the query point.



(a) (b)

Figure 32: Data space configuration of *Lung* data set after applying WF method (a) entire space (b) 50 nearest elements from the query point.



(a) (b)

Figure 33: Data space configuration of *Lung* data set after applying TF method (a) entire space (b) 50 nearest elements from the query point.

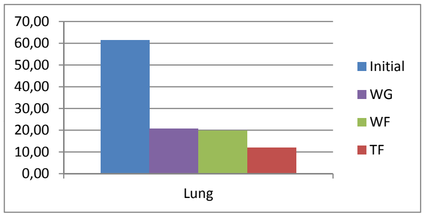


Figure 34: MSD values of clusters of relevant elements on *Lung* data set.

for the lung data set because it can capture the peculiarities of lung images, specially regarding well-defined contours.

Figures 31(b), 32(b), 33(b) show the space with only the 50 closest elements. In the initial space, the number of relevant elements retrieved in the first 50 closest elements retrieved was 13 (Figure 30(b)). The best result after optimization was obtained by TF, with 31 relevant elements, followed by WF with 22 and WG with 18 relevant elements retrieved. Figure 34 shows that, once more, TF had the best cohesion on the relevant elements, and WF was slightly better than WG. It can be observed that all optimization methods generated clusters of relevant elements significantly superior in comparison to the initial space.

ACKNOWLEDGMENTS

The authors are grateful for the financial support granted by CNPq, FAPESP, CAPES, and Microsoft Research.

6. CONCLUSIONS

In this study we have employed Genetic Algorithm techniques combined with Relevance Feedback techniques to improve the accuracy of CBIR systems; we followed this course of action according to two different novel approaches. The first one (WF) infers a weight vector to adjust the Euclidean dissimilarity function by means of weighting functions; the second one (TF) aims at optimizing the features space using linear and non linear transformation functions.

We performed several experiments using images from a general domain and from the medical domain. The results considering feature space transformation functions achieved successful effectiveness, obtaining the highest precision in all experiments, outperforming the other methods by up to 70%. The higher performance obtained by using the features space transformation is due to the fact that it leads to a configuration in which the space is transformed by different functions such as polynomials with several different degrees. Meanwhile, weighted distance functions limit the configuration to linear transformations.

The weighting function approach was more effective than the simple weighting in continuous interval. The advantage of using weighting functions instead of directly generate weights is that the search space is considerably reduced.

We have used visual data analysis in order to visualize the features spaces so to demonstrate that method TF generated space configurations that better express the semantic related to the user's interests; in such visualizations, the clusters of the relevant elements of the queries were grouped closer to the center, as expected. As future work, one possible approach is to investigate the use of the weighting and transformation functions for further feature extractors and distance functions, as well as to study possible correlations between the different feature extractors and their behavior.

7. REFERENCES

[1] Y. Alemu, J. bin Koh, M. Ikram, and D.-K. Kim. Image retrieval in multimedia databases: A survey. In *Proceedings of the Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, pages 681–689, Los Alamitos, CA, USA, 2009.

[2] S. Brandt. Use of shape features in content-based image retrieval. Master's thesis, Helsinki University of Technology, 1999.

[3] P. Bugatti, A. Traina, and C. Traina. Improving content-based retrieval of medical images through dynamic distance on relevance feedback. In *Proceedings 24th International Symposium on Computer-Based Medical Systems(CBMS)*, pages 1–6, Bristol, IK, 2011.

[4] P. H. Bugatti, A. J. M. Traina, and C. Traina-Jr. Assessing the best integration between

distance-function and image-feature to answer similarity queries. In *Proceedings of the 2008 ACM Symposium on Applied Computing (SAC)*, pages 1225–1230, Fortaleza, Ceara, Brazil, 2008.

[5] S. F. da Silva, M. X. Ribeiro, J. do E.S. Batista Neto, C. Traina-Jr., and A. J. Traina. Improving the ranking quality of medical image retrieval using a genetic feature selection method. *Decision Support Systems*, 51(4):810–820, 2011.

[6] J. dos Santos, C. Ferreira, R. da S. Torres, M. Gonçalves, and R. Lamparelli. A relevance feedback method based on genetic programming for classification of remote sensing images. *Information Sciences*, 181(13):2671 – 2684, 2011.

[7] R. M. Haralick, K. Shanmugam, and I. Dinstein. Textural Features for Image Classification. *IEEE Transactions on Systems, Man and Cybernetics*, 3(6):610–621, 1973.

[8] A. Lakdashti, M. Shahram Moin, and K. Badie. Semantic-based image retrieval: A fuzzy modeling approach. In *Proceedings of the IEEE/ACS International Conference on Computer Systems and Applications*, pages 575–581, Doha, Qatar, 2008.

[9] J. Li and J. Z. Wang. Automatic linguistic indexing of pictures by a statistical modeling approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25:1075–1088, 2003.

[10] H.-H. Loh, J.-G. Leu, and R. Luo. The analysis of natural textures using run length features. *IEEE Transactions on Industrial Electronics*, 35(2):323 –328, 1988.

[11] C. López-Pujalte, V. P. Guerrero-Bote, and F. de Moya-Anegón. Order-based fitness functions for genetic algorithms applied to relevance feedback. *Journal of the American Society for Information Science and Technology*, 54(2):152–160, January 2003.

[12] D. G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the 7th IEEE International Conference on Computer Vision*, volume 2, pages 1150–1157, Los Alamitos, CA, USA, 1999.

[13] d. l. T. F. Nguyen, M.H. Optimal feature selection for support vector machines. *Pattern Recognition*, 43:584–591, 2010.

[14] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision*, 42:145–175, 2001.

[15] M. Ribeiro, A. Balan, J. Felipe, A. Traina, and C. Traina. Mining statistical association rules to select the most relevant medical image features. In D. Zighed, S. Tsumoto, Z. Ras, and H. Hacid, editors, *Mining Complex Data*, volume 165 of *Studies in Computational Intelligence*, pages 113–131. Springer Berlin / Heidelberg, 2009.

[16] J. F. Rodrigues Jr., L. A. M. Zaina, L. A. S. Romani, and R. R. Ciferri. Metricsplat - a platform for quick development, testing and visualization of content-based retrieval techniques. In *Proceedings of the 24th Brazilian Symposium on Databases*, Fortaleza, Ceara, Brazil.

[17] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12):1349–1380, 2000.

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

[19] M. A. Stricker and M. Orengo. Similarity of color images. In *Proceedings of the Storage and Retrieval for Image and Video Databases (SPIE*, pages 381–392, San Diego CA, USA, 1995.

[20] J.-H. Su, W.-J. Huang, P. S. Yu, and V. S. Tseng. Efficient relevance feedback for content-based image retrieval by mining user navigation patterns. *IEEE Transactions on Knowledge and Data Engineering*, 23(3):360–372, 2011.

[21] R. d. S. Torres, A. X. Falcão, M. A. Gonçalves, J. a. P. Papa, B. Zhang, W. Fan, and E. A. Fox. A genetic programming framework for content-based image retrieval. *Pattern Recognition*, 42:283–292, February 2009.

[22] P. Zezula, G. Amato, V. Dohnal, and M. Batko. *Similarity Search: The Metric Space Approach*, volume 32 of *Advances in Database Systems*. Springer, 2006.

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

ABOUT THE AUTHORS:

Letricia P. S. Avalhais received the MSc degree in Computer Science and Computational Mathematics at University of Sao Paulo (in January 2012), and the BSc in Computer Science at Federal University of Mato Grosso do Sul (in December 2008). She is currently a Ph.D. candidate at University of Sao Paulo. Her research interests include Image Processing, Content-Based Image Retrieval, Video Analysis, Visualization and Machine Learning.

Sergio F. Silva received the B.Sc. degree in computer science from the Federal University of Goias, Brazil, in 2004. He received the the M.Sc. degree in computer science at the Faculty of Computation of the University of Uberlandia, Brazil, in 2007 and Ph.D. degree in computer science at the Mathematics and Computer Science Institute, University of Sao Paulo at Sao Carlos, Brazil. His research interests include multimedia data mining, computer-aided diagnosis and content-based image retrieval with special attention for optimization techniques based on evolutionary computation.

Jose F. Rodrigues Jr. is a Professor at University of Sao Paulo, Brazil. He received his Ph.D. from this same university, part of which was carried out at Carnegie Mellon University in 2007. Jose Fernando is a regular author and reviewer of major events in his field, having contributed with publications in IEEE and ACM journals and conferences. His topics of research include data analysis, content-based data retrieval, and visualization.

Agma J. M. Traina received the B.Sc., the M.Sc. and Ph.D. degrees in computer science from the University of Sao Paulo, Brazil, in 1983, 1987 and 1991, respectively. She is currently a full Professor with the Computer Science Department of the University of Sao Paulo at Sao Carlos, Brazil. Her research interests include image databases, image mining, indexing methods for multidimensional data, information visualization, image processing for medical applications and optimization techniques based on evolutionary computation.

Caetano Traina Jr. received the B.Sc. degree in electrical engineering, the M.Sc. and Ph.D. degrees in computer science from the University of Sao Paulo, Brazil, in 1978, 1982 and 1987, respectively. He is currently a full professor with the Computer Science Department of the University of Sao Paulo at Sao Carlos, Brazil. His research interests include access methods for complex data, data mining, similarity searching and multimedia databases.