Preprint version of paper published in Information Sciences journal, 2012.
DOI: http://doi.org/10.1016/j.ins.2012.04.032

Exploiting Pairwise Recommendation and Clustering Strategies for Image Re-Ranking

Daniel Carlos Guimarães Pedronette and Ricardo da S. Torres

RECOD Lab - Institute of Computing (IC)
University of Campinas (UNICAMP)
Campinas, Brazil
dcarlos@ic.unicamp.br, rtorres@ic.unicamp.br

Abstract

In Content-based Image Retrieval (CBIR) systems, accurately ranking collection images is of great relevance. Users are interested in the returned images placed at the first positions, which usually are the most relevant ones. Commonly, image content descriptors are used to compute ranked lists in CBIR systems. In general, these systems perform only pairwise image analysis, that is, compute similarity measures considering only pairs of images, ignoring the rich information encoded in the relations among several images. This paper presents a novel re-ranking approach used to improve the effectiveness of CBIR tasks by exploring relations among images. In our approach, a *recommendation-based strategy* is combined with a clustering method. Both exploit contextual information encoded in ranked lists computed by CBIR systems. We conduct several experiments to evaluate the proposed method. Our experiments consider shape, color, and texture descriptors and comparisons with other post-processing methods. Experimental results demonstrate the effectiveness of our method.

Keywords: content-based image retrieval, re-ranking, rank aggregation, recommendation

1. Introduction

Traditional image retrieval approaches, based on keywords and textual metadata, face serious challenges [9]. Describing the image content with textual features is intrinsically very difficult, and this task has become even harder due to the growth and diversification of image collections. In many applications, especially those dealing with large and heterogeneous image collections, there are several obstacles to define appropriate textual descriptors: the manual annotation is prohibitively expensive, contextual text is scarce or unreliable, and user needs are impossible to anticipate.

One of the commonest approaches to overcome the limitations of manually describing the image content relies on the use of Content-Based Image Retrieval (CBIR) systems. Many of these systems are based on image features which can be computed directly and automatically from the images themselves [6]. Basically, given a query image, a CBIR system aims at retrieving the most similar images in a collection by taking into account image visual properties (such as, shape, color, and texture). Collection images are *ranked* in decreasing order of similarity, according to a given *image descriptor* or to a set of image descriptors.

In the past few years, several CBIR approaches have been proposed considering applications on different areas, from facial image retrieval [38] to remote sensing images [31]. However, in general, CBIR image descriptors perform only pairwise image analysis and compute similarity (or distance) measures considering only pairs of images, ignoring the rich information encoded in the relations among several images [21].

In this paper, we present a new re-ranking method that takes into account relationships among images for improving the effectiveness of CBIR descriptors. We propose a measure for analysing the quality of ranked lists and the use of the concept of *recommendation* for establishing new relationships among images, given identified high-quality ranked lists. Recommender systems attempt to reduce information overload by selecting automatically items that match the personal preferences of each user [4, 33]. More formally, "given a collection and an actor, and a set of ratings for objects in that collection produced by others or the same actor, recommends (produces a subset of that collection) for that particular actor [10]".

Our pairwise recommendation approach is inspired by the concept of recommendation, originally created to consider user ratings. However, our method does not require any user interaction. The recommendations are simulated based on contextual information encoded in ranked lists computed by CBIR descriptors. The relationships among images are used for composing *image profiles* and then recommending images, i.e., an image is able to recommend images (that are possibly relevant) to another image. In this context, a recommendation means that the distance between two images should be decreased and an image should be *moved up* in the ranked list of the image that received the recommendation.

Our method also incorporates a simple clustering step method with the objective of further improving distances among images that belong to a same cluster. Furthermore, our approach can also be used for combining different CBIR descriptors (rank aggregation tasks).

We conducted a large evaluation protocol involving shape, color, and texture descriptors datasets and comparisons with other post-processing approaches. Experimental results demonstrate the effectiveness of our method. The re-ranking algorithm yields better results in terms of effectiveness performance than various post-processing algorithms recently proposed in the literature [13, 22, 42, 43].

This paper is organized as follows. Section 2 discusses related work. Section 3 presents the image re-ranking algorithm based on pairwise recommendation. Section 4 describes the experimental evaluation and, finally, Section 5 presents our conclusions.

2. Related Work

Several methods have been proposed to perform re-ranking tasks on various information retrieval systems [2, 7, 12, 21, 25, 27, 32, 37]. In a general way, these *post-processing methods* take an initial ranking and use additional information (relationship among items, user preferences, or other rankings) for improving the effectiveness of the retrieval process. In this section, we aim at briefly discussing some approaches.

A definition for the term *"global ranking"* was proposed in [27] and was used in Information Retrieval tasks. Basically, a global ranking approach considers that relations always exist between objects and it is better to define the ranking model as a function of all the objects to be ranked. In [12], an approach that explores information of user clicks was proposed for re-ranking in a web search scenario. Inter-document similarity are considered in [7] and a clustering approach is used to regularize retrieval scores. In [41], a semi-supervised label propagation algorithm [44] was used for re-ranking documents in information retrieval applications.

In the CBIR domain, the concept of *"contextual information"* has been used for designing methods that take information about relationships among images for re-ranking. In [25], the notion of context refers to nearest neighbors of a query. A similarity measure especially proposed for ranked lists is employed to characterize contextual information. An extension of this approach was proposed in [32]. A clustering method is considered for representing the context information. In [21], gray scale images are used for representing contextual information. This approach applies image processing techniques for handling contextual information used for re-ranking.

The RL-Sim Re-Ranking Algorithm [24] was proposed considering similarity between ranked lists for characterizing contextual information. The main motivation of the algorithm relies on the conjecture that contextual information encoded in the similarity between ranked lists can provide useful information for improving the effectiveness of CBIR descriptors. The Distance Optimization Algorithm (DOA) is presented in [23]. DOA considers an iterative clustering approach based on distances correlation and on the similarity of ranked lists. The algorithm explores the fact that if two images are similar, their distances to other images and therefore their ranked lists should be similar as well.

Various methods have also been proposed for post-processing shape matching tasks, considering relationships among all shapes. A graph transduction learning approach is introduced in [42]. The algorithm computes the similarity of a pair of shapes in the context of other shapes as opposed to considering only pairwise relations. This method is an application of semi-supervised label propagation algorithm [44]. The influence among shape similarities in an image collection is analyzed in [43]. Markov chains are used to perform a diffusion process on a graph formed by a set of shapes, where the influences of other shapes are propagated. The approach introduces a locally constrained diffusion process and a method for densifying the shape space by adding synthetic points. A method that exploits the shape similarity scores is proposed in [13]. This method uses an unsupervised clustering algorithm, aiming at capturing the manifold structure of the image relations by defining a neighborhood for each data point in terms of a mutual k-nearest neighbor graph. In [22], a distance optimization algorithm has been proposed. The objective is to cluster shapes by taking into account the similarity among ranked lists. Distances between shapes are updated based on created clusters aiming at improving the retrieval effectiveness.

Several methods refer to the same concepts using different terminologies (e.g., *global ranking, contextual information, and diffusion process*). However, the main idea of all methods is very similar: given an initial set of ranked lists, additional information (e.g., user clicks, relationship among objects) is considered in the ranking process aiming at improving the effectiveness of the CBIR systems.

This paper presents a novel re-ranking method for CBIR systems, which exploits the notion of *recommendation* for modeling and handling relationships among images. The proposed approach based on pairwise recommendation is the main novelty of this paper, which is conceptually very different from previous works [21, 23, 24]. We believe that our strategy opens a new area of investigation related to the use of recommendation techniques in re-ranking tasks. Our method is detailed in the following section.

3. Re-Ranking Algorithm

This section presents the proposed re-ranking algorithm.

3.1. Problem Definition

Let $\mathcal{C}=\{img_1, img_2, \ldots, img_N\}$ be an *image collection.*

Let $\mathcal{D}$ be an *image descriptor* which can be defined [29] as a tuple (ϵ, ρ), where:

- ϵ: $\hat{I} \rightarrow \mathbb{R}^n$ *is a function that extracts a feature vector* $v_{\hat{I}}$ *from an image* $\hat{I}$.
- ρ: $\mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$ *is a distance function that computes the distance between two images as a function of the distance between their corresponding feature vectors.*

In order to obtain the distance between two images img_i and img_j it is necessary to compute the value of $\rho(\epsilon(img_i),\epsilon(img_j))$. For simplicity and readability purposes, we use the notation $\rho(img_i,img_j)$ along the paper.

The distance $\rho(img_i,img_j)$ among all images $img_i,img_j \in \mathcal{C}$ can be computed to obtain an $N \times N$ distance matrix A.

Given an image query img_q, we can compute a ranked list R_q in response to the query, by taking into account the distance matrix A. The ranked list $R_q=\{img_1, img_2, \ldots, img_N\}$ can be defined as a permutation of collection $\mathcal{C}$, such that, if img_1 is ranked before img_2 (top positions), then $\rho(img_q,img_1) \leq \rho(img_q,img_2)$. We can also take every image $img_i \in \mathcal{C}$ as a query image img_q, in order to obtain a set $\mathcal{R} = \{R_{img_1}, R_{img_2}, \ldots, R_{img_N}\}$ of ranked lists for each

Table 1: Meaning of Symbols

Symbol	**Meaning**
$\mathcal{C}$	Image collection.
$\mathcal{D}$	Image descriptor.
ρ	Image descriptor distance function.
N	Size of collection.
A	Initial distance matrix.
$\mathcal{R}$	Initial set of ranked lists.
$\hat{A}$	Distance matrix after re-ranking.
$\hat{\mathcal{R}}$	Set of ranked lists after re-ranking.

image of collection $\mathcal{C}$. Our goal is to propose a re-ranking algorithm (represented by function f) that takes as input the distance matrix A and the set of ranked lists $\mathcal{R}$ for computing a new distance matrix $\hat{A}$:

$$\hat{A} = f(A, \mathcal{R}) \tag{1}$$

Given the new distance matrix $\hat{A}$, a new set $\hat{\mathcal{R}}$ can be obtained. $\hat{\mathcal{R}}$ contains the new ranking positions of all collection images, that is, the collection images are re-ranked. Note that the main aspect of f consists in exploiting all relationship information encoded in A and $\mathcal{R}$. The definition of function f in a CBIR scenario is similar to the concept of *global ranking* [27] used in information retrieval domain.

Table 1 summarizes the used symbols and their respective meanings.

3.2. The Re-Ranking Algorithm

The main idea of our re-ranking algorithm relies on the conjecture that images can *recommend* images found at the first positions of their ranked lists (that is, their *K-nearest-neighbors*). In this scenario, *recommendation* means decreasing the distance between images: when an image img_i recommends an img_k to an image img_j, it means that image img_j should have its distance to img_k decreased.

Each recommendation is associated with a different *weight* (how much the distance should be decreased). For computing the recommendation weight, we consider the position of images in ranked lists and the *quality* of the ranked lists. We use a *cohesion* measure for estimating the quality of ranked lists and then sorting the ranked lists. We consider, in this way, that images with better ranked lists (higher cohesion) have more authority for making recommendations. After performing all recommendations, ranked lists are considered for clustering images and additional recommendations are made given the obtained clusters.

Once all distances have been updated by recommendations, a re-ranking can be performed based on the new distance matrix A_{t+1} (where t indicates the current iteration) for generating a new set of ranked lists $\mathcal{R}_{t+1}$. These steps are repeated in an iterative manner until a convergence criterion is reached.

The employed convergence criterion is based on the variation of the cohesion measure. At each iteration, we increment the number K of neighbors considered for recommendations. Note that after one iteration, more relevant images are found at first positions of the ranked lists. Non-relevant images are moved out from the first positions of the ranked lists and therefore K can be increased for considering more images. In the next iteration, more images (larger K) are considered in the recommendation process. Finally, when the convergence criterion is reached, a re-ranking is performed based on the final distance matrix $\hat{A}$. Figure 1 illustrates the main steps of our approach. Algorithm 1 outlines our re-ranking method.

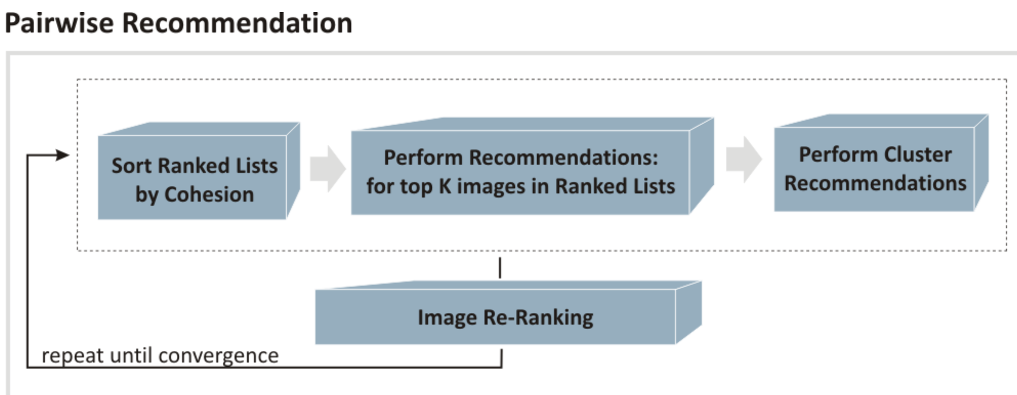

Figure 1: Pairwise Recommendation re-ranking method.

The main steps of Algorithm 1 are presented in Lines 6, 10, and 13, which refer, respectively, to computing the *cohesion*, to making *recommendations*, and to *clustering* images. These steps are detailed in the next sub-sections. Note that, in Line 8, $C = \{c_1, c_2, \ldots, c_N\}$ is a set of cohesion scores c_i computed for each ranked list R_i. Based on C, a set $\mathcal{R}_c$ is computed, where ranked lists are sorted in decreasing order of cohesion. In Line 16, a re-ranking is performed. Once the distance matrix A_{t+1} is updated, the ranked lists are computed again, that is, images are re-ranked.

3.3. Cohesion Measure

In this paper, we use a *cohesion* measure for estimating the quality of *ranked lists*. The objective of this measure is to assess how "good" a ranked list is. A ranked list is considered "good" when images placed at the top positions refer to each other at the top positions of their ranked lists. It is somehow close to the cluster hypothesis [28], which states that *"closely associated documents tend to be relevant to the same requests"*.

Our method considers that *"high quality"* ranked lists are able to make more accurate recommendations. In this sense, these ranked lists have more authority (defined by the cohesion measure) to make recommendations. This approach

Algorithm 1 Pairwise Recommendation Re-Ranking

Require: Distance matrix A and set of ranked lists $\mathcal{R}$

Ensure: New distance matrix $\hat{A}$ and new set $\hat{\mathcal{R}}$

$t \leftarrow 0$
$A_t \leftarrow A$
$currentCohesion \leftarrow 0$
$\mathcal{R}_t \leftarrow \mathcal{R}$
repeat
  for all $R_i \in \mathcal{R}$ **do**
    $c_i \leftarrow computeCohesion(R_i, \mathcal{R}_t)$
  end for
  $\mathcal{R}_c = sortRankedListsByCohesion(\mathcal{R}_t, C)$
  for all $R_i \in \mathcal{R}_c$ **do**
    $performRecommendations(A_t, R_i, c_i)$
  end for
  for all $R_i \in \mathcal{R}_c$ **do**
    $performClusterRecommendations(A_t, R_i)$
  end for
  $A_{t+1} \leftarrow A_t$
  $\mathcal{R}_{t+1} \leftarrow performReRanking(A_t)$
  $lastCohesion \leftarrow currentCohesion$
  $currentCohesion \leftarrow computeAvgCohesion(R_{t+1})$
  $t = t + 1$
  $K = K + 1$
until $(currentCohesion - lastCohesion) < (currentCohesion \times \epsilon_{cohesion})$
$\hat{A} = A_t$
$\hat{\mathcal{R}} = \mathcal{R}_t$

is analogous to the PageRank algorithm [19]. Although having different objectives, both PageRank and our cohesion measure exploit the link structure (hyperlinks/references in ranked lists) for obtaining information about items (pages/images). Basically, the PageRank algorithm assesses the importance of a page by taking into account link structures. In our approach, the cohesion measure aims at assessing the quality of ranked lists by analyzing how images refer to each other in their ranked lists.

The value of the measure is normalized in the interval [0,1], where value 1 indicates the highest possible cohesion. Let R_i be a ranked list of an image img_i. Let R_{ki}=$\{img_1, img_2, \ldots, img_K\}$ be a subset of a ranked list R_i that considers the K nearest neighbors of img_i. Let $img_j \in R_{ki}$ be an image of this subset (one of K-neighbors of image img_i), and let R_{kj} be a subset of the ranked list of img_j. Finally, let $img_p \in R_{kj}$ be an image at position p of the ranked list R_{kj}. We define the cohesion as follows:

$$cohesion(R_i, K) = \frac{\sum_{img_j \in R_{ki}} \sum_{img_p \in R_{kj}} w(p) \times s(R_{ki}, img_p)}{\sum_{img_j \in R_{ki}} \sum_{img_p \in R_{kj}} w(p)} \tag{2}$$

The terms s and w are functions. The objective of function s is to determine if image img_p (that belongs to subset R_{kj}) also belongs to subset R_{ki}. The function s is defined as follows:

$$s(R_{ki}, img_p) = \begin{cases} 1, & \text{if } img_p \in R_{ki} \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

The function w takes as input a position of an image in a ranked list. The goal is to give high weights to images at the first positions of ranked lists. In our algorithm, we define w as $w(p) = 1/p$. Note that, if all referenced images are in the subset R_{ki}, the function s will assume value 1 for all images and therefore cohesion (Equation 2) is set to 1. It indicates a perfect cohesion, where all considered images refer to each other at the first positions of their ranked lists.

Figure 2 illustrates the computation of the cohesion measure for the ranked list R_i. Observe, on the left, the ranked list R_i and its subset R_{ki}. On the right, for a given image $img_j \in R_{ki}$, it illustrates the ranked list R_j (and its subset R_{kj}). The function s verifies if an img_p belongs to both subsets R_{ki} and R_{kj}. Function w, illustrated on the right, computes a weight given the position of image img_p in the ranked list R_j.

3.4. Performing Recommendations

The basic idea of our recommendation method is: "*an image img_i recommends the img_y to img_x, if img_x and img_y are at the top-K positions of the ranked list of img_i*". In this context, the *recommendation* is associated with a *decrease of the distances* between two images (img_x and img_y). The recommendations are performed using the same information considered in the computation of the cohesion of ranked lists: a subset R_{ki} with the K-nearest neighbors in the ranked list R_i. Observe that, before recommendations, the cohesion of all

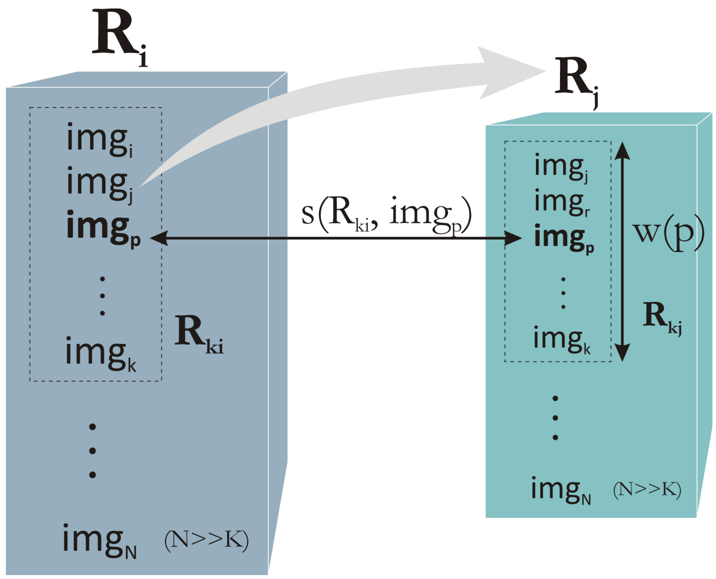

Figure 2: Computation of the cohesion measure.

ranked lists were computed and the ranked lists were sorted in a decreasing order of cohesion. In this way, the recommendations, which represent updates for distance matrix A, will be performed first for ranked lists with higher cohesion. Algorithm 2 presents our method for performing recommendations for a given ranked list R_i.

Variables w_x and w_y represent the weight given to images img_x and img_y in the recommendation. The weights are computed based on the position of the images in the ranked lists: for images at first positions of a ranked list, a higher weight is assigned. The weights associated with the first positions indicate where it is more likely to find the most similar (relevant) images, that is, positions that represent more reliable recommendations. These variables are computed in Lines 4 and 7 of Algorithm 2, both in the interval [0,1]. In Line 8, the weight w of a recommendation is computed. That represents the *reputation* of the recommendation. For computing w, we consider w_x, w_y and the cohesion c_i of the ranked list R_i. Figure 3 illustrates how a recommendation is performed for a given ranked list R_i. It considers two images img_x, $img_y \in R_{ki}$ and takes into account their positions in the ranked list for computing the weights w_x and w_y.

In Line 9, a coefficient λ is computed in the interval [0,1]. This coefficient is used to determine how the distances between img_x and img_y should be decreased. For computing λ, we multiply the weight w of the recommendation and a constant L. The goal of constant L is to adjust the *"speed"* of the convergence of the algorithm. By increasing the value of L, the distances among images will decrease faster and the algorithm will be executed in less iterations.

Algorithm 2 Making Recommendations

Require: Matrix A, Ranked list R_i and Cohesion c_i
Ensure: Updated matrix A

1: $R_{ki} \leftarrow KNN(R_i)$
2: $x \leftarrow 1$
3: **for all** $img_x \in R_{ki}$ **do**
4: $\quad w_x \leftarrow 1 - (x/K)$
5: $\quad y \leftarrow 1$
6: $\quad$ **for all** $img_y \in R_{ki}$ **do**
7: $\quad\quad w_y \leftarrow 1 - (y/K)$
8: $\quad\quad w \leftarrow c_i \times w_x \times w_y$
9: $\quad\quad \lambda \leftarrow 1 - min(1, L \times w)$
10: $\quad\quad A[x, y] \leftarrow min(\lambda A[x, y], A[y, x])$
11: $\quad\quad y \leftarrow y + 1$
12: $\quad$ **end for**
13: $\quad x \leftarrow x + 1$
14: **end for**

However, with a very high value of L [1], the algorithm cannot take advantage of the improvements of the ranked lists along iterations. Note also that we use a *min* function in Line 9 to avoid negative values for λ. Finally, the value of λ is multiplied by the current distance $A[x, y]$ for computing the new updated distance.

3.5. Clustering Approach

High values of w (or L) can lead to situations where $\lambda = 0$ and, consequently, $A[x, y]$=0. These cases are associated with recommendations of great confidence. The key idea of our clustering approach is to exploit these cases to group images and then making additional recommendations based on created clusters. Let R_i be a ranked list of an image img_i. A cluster Cl_i is composed by all images whose distance to img_i is equal to 0. Cl_i can also be defined as follows: $\{Cl_i \subset R_i \mid \forall\ img_c \in Cl_i,\ A[i, c] = 0\}$.

Given a cluster Cl_i the additional recommendations consists in setting all distances among all images of C_i to 0. More formally: we aim at ensuring that for each cluster Cl_i and for each pair of images $img_x, img_y \in Cl_i$, we have $A[x, y] = A[y, x] = 0$.

3.6. Convergence Criterion

In general, an iterative method is said to converge, if the difference between results obtained along iterations decreases, tending to reach an ultimate result. In our case, it is expected that the proposed re-ranking algorithm converges, improving the quality of the ranked lists along the iterations.

[1] We used L in the interval [1,2] in our experiments.

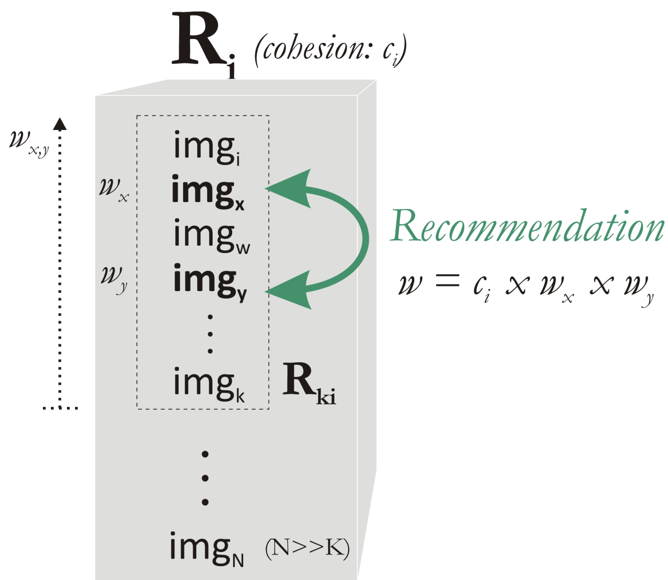

Figure 3: Making recommendations.

In Section 3.3, we described the *cohesion* measure, whose main goal is to estimate the quality of ranked lists. This measure is also used as a convergence criterion, according the follow conjecture: *"the re-ranking procedure should be iteratively executed while the quality of ranked lists (measured by cohesion) is increasing"*. Therefore, at each iteration, the average cohesion of all ranked lists is computed and compared with the one computed in the previous iteration. The convergence criterion of the re-ranking algorithm is tested in Line 22 of Algorithm 1. The convergence condition checks if the variation of cohesion is greater than a given threshold. The threshold is computed proportionally to the current cohesion, using the parameter $\epsilon_{cohesion}$. For the convergence criterion, the computation of cohesion considers the $2 \times K$ top positions of ranked lists (initial value of K).

In the following, we present a brief discussion about the method convergence. Let $\mathcal{C}$ be an image collection. Let S_i be a set of similar images such that $\{S_1 \cup S_2 \cup \cdots \cup S_m\} = \mathcal{C}$ and $\mid S_i \mid \geq K$. We consider three hypothetical scenarios, given the effectiveness of used CBIR descriptors:

1. *"Highly effective descriptor":* by using the most effective descriptor for collection $\mathcal{C}$, images found at the top K positions of a ranked list R_{ki} of an image $img_i \in S_i$ are all similar to each other, that is $R_{ki} \subset S_i$. In this scenario, the average cohesion of ranked lists is very high, since all similar images refer to each other at the top positions of their ranked lists. Therefore, the recommendations produce small changes in the ranked lists. In this way, the variation of average cohesion is very low and the convergence is reached very quickly.

2. *"Real-world descriptor":* for a real-word descriptor, the ranked list R_{ki} may include some incorrect results, that is, some non-similar images are found at the top K positions of R_{ki}. Let $img_j \in R_{ki}$ be an image non-similar to img_i. In that case, recommendations defined for ranked lists of similar images to img_i can improve R_{ki}, by moving the non-similar image img_j out of the first positions of R_{ki}. In other words, when correct results represent the common case, the recommendation method can improve ranked lists. While these improvements occur, the average cohesion of ranked lists increases. That process is repeated until convergence is reached.
3. *"Non-effective descriptor":* for non-effective descriptors, the created ranked lists can be seen as a result of a random permutation of images. In that case, the method convergence would be slow as the number of similar images found at the top positions of ranked lists are very small.

An experimental analysis of convergence is presented in Section 4.4.

3.7. Re-Ranking for Rank Aggregation

Recently, several methods have been proposed aiming at combining ranked lists produced by different descriptors. The objective is to produce better effectiveness results [2, 9, 37]. We aim at proposing an application of our re-ranking algorithm for combining descriptors (rank aggregation). Let $\mathcal{C}$ be an image collection and let $\mathcal{D} = \{D_1, D_2, \ldots, D_m\}$ be a set of CBIR descriptors. We can use the set of descriptors $\mathcal{D}$ for computing a set of distance matrices $\mathcal{A} = \{A_1, A_2, \ldots, A_m\}$. Our approach to combine descriptors works as follows. The first step is to combine the set $\mathcal{A}$ in a unique matrix A_c. For the matrices combination, we use a multiplicative approach. Every (i, j) position of matrix is computed as follows:

$$A_c[i,j] = A_1[i,j] \times A_2[i,j] \times \cdots \times A_m[i,j] \tag{4}$$

Once we have a matrix A_c, we compute a set of ranked lists $\mathcal{R}_c$ based on this matrix. Then, we perform our re-ranking algorithm now using the matrix A_c and the set $\mathcal{R}_c$.

4. Experimental Evaluation

In this section, we present a set of conducted experiments for demonstrating the effectiveness of our method. We analyzed and compared our method under several aspects. Section 4.1 discusses the impact of different parameters of the proposed method in terms of effectiveness and efficiency.

Section 4.2 presents results related to the use of our method with several shape descriptors, considering the well-known MPEG-7 dataset [15]. Sections 4.3 aims at validating the hypothesis that our method can be used in general image retrieval tasks. In addition to shape descriptors, we conduct experiments with color and texture descriptors. The objective of the experiments presented in these sections is to assess the effectiveness of the method considering different visual properties and different datasets. Section 4.4 discusses convergence aspects of the re-ranking method.

We also conducted experiments with the objective of comparing our results with state-of-the-art post-processing methods in Section 4.5. Our comparison considers shape descriptors and two datasets: MPEG-7 and Kimia-99. Finally, Section 4.6 presents experimental results of our re-ranking method when used to combine descriptors. We conduct experiments for shape, color, and texture descriptors. We also compare our re-ranking algorithm with other combination methods.

All experiments were conducted considering all images in the collections as query images. Results presented (in terms of MAP and Recall@40 scores) represent the average score considering all queries.

4.1. Impact of Parameters

The execution of Algorithm 1 considers three parameters: *(i)* K - number of initial neighbors considered for recommendations; *(ii)* L - a constant that controls the influence of weights; and *(ii)* $\epsilon_{cohesion}$ - the threshold parameter considered in the convergence criterion computation (basically, it determines the number of iterations along which the algorithm is executed). To evaluate the influence of different parameter settings on the retrieval scores and to determine the best parameters values, we conducted a set of experiments. We use MPEG-7 dataset [15]. The MPEG-7 dataset is a well-known shape database, composed by 1400 shapes divided into 70 classes. The size of images ranges from 50×48 to 526×408 pixels. For evaluation, the so-called bullseye score was considered, which counts all matching objects within the 40 most similar candidates. Since each class consists of 20 objects, the retrieved score is normalized with the highest possible number of hits. For distance computation, we used the CFD [22] shape descriptor. Retrieval scores are computed ranging parameters K in the interval [1,15] and T in the interval [1,30] (with increments of 5) for each value of L.

Figures 4 and 5 show surfaces that represent retrieval scores for L equal to 1 and 2, respectively. We can observe optimal combinations of values for regions close to $K = 8$ and $T = 15$, for which the best retrieval scores are observed. In the following experiments, we use $K = 8$ and $\epsilon_{cohesion} = 0.0125$ (threshold that reaches convergence in about 15 iterations). Note that although these parameters were defined considering a single descriptor/dataset, they were used in all conducted experiments with good results [2].

Figure 6 shows the impact of different values of L in the method precision. We set $K = 8$ and $T = 15$ and computed the retrieval scores for L in the interval [0,3]. In this case, the best retrieval score was reached for $L = 2$. The value of $L = 2$ indicates that a high weight can be assigned to the recommendations, that

[2] These parameters may change for database with very different sizes.

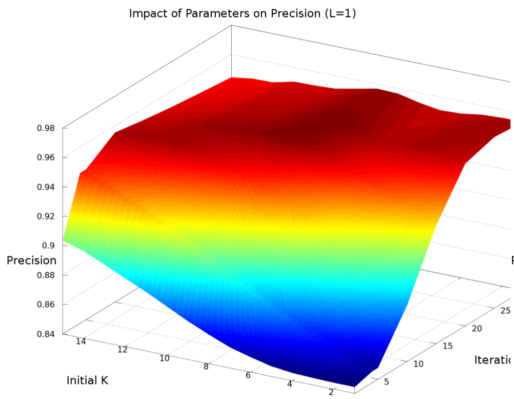

Figure 4: Impact of parameters K and T for $L = 1$.

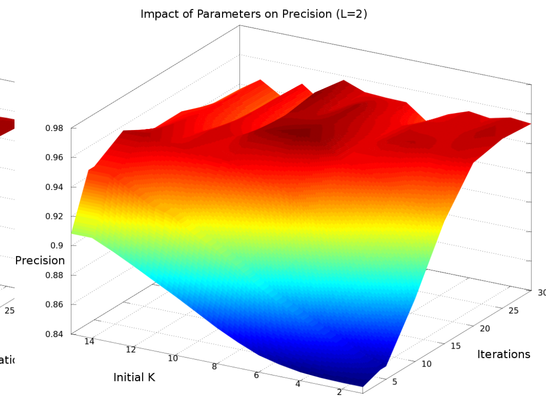

Figure 5: Impact of parameters K and T for $L = 2$.

increases their strength and impacts positively the effectiveness of the proposed method.

Finally, we analyze the impact of parameters K and T on computation time. Figure 7 illustrates a surface representing the variation of computation time as a function of K and T. We can observe a quadratic behavior for the surface. The computation time increases proportionally to $K^2 \times T$. Note that, although the computation time is quadratic for parameters K and T, it is linear ($O(N)$) for the size of collection N, since the recommendations are considered for $K << N$ images.

4.2. Shape Descriptors

We evaluate the use of our method with four shape descriptors: Segment Saliences (SS) [30], Beam Angle Statistics (BAS) [1], Inner Distance Shape Context (IDSC) [16], and Contour Features Descriptor (CFD) [22]. This experiment considers the MPEG-7 dataset and the bullseye score. Parameters are set according to experimental analysis presented in the previous section: $K = 8$, $\epsilon_{cohesion} = 0.0125$, and $L = 2$. Figure 8 illustrates an example of results for a MPEG-7 shape taken as query. The comparison considers the CFD [22] shape descriptor before and after the use of the proposed re-ranking method. Results of bullseye score for all descriptors are presented in Table 2. Note that the effectiveness gains are always positive and represent very significant improvement of effectiveness, ranging from +7.97% to +23.57%. In Figure 9, we report the percentage gains obtained by using the Pairwise Recommendation algorithm for each of 70 shape classes in the MPEG-7 dataset. Note that the bullseye score was improved by over 10% on average, and over 50% for two classes.

4.3. General CBIR Tasks

In general, post-processing methods [13, 20, 22, 42, 43] have been evaluated their approaches for only one type of visual property (usually, either color

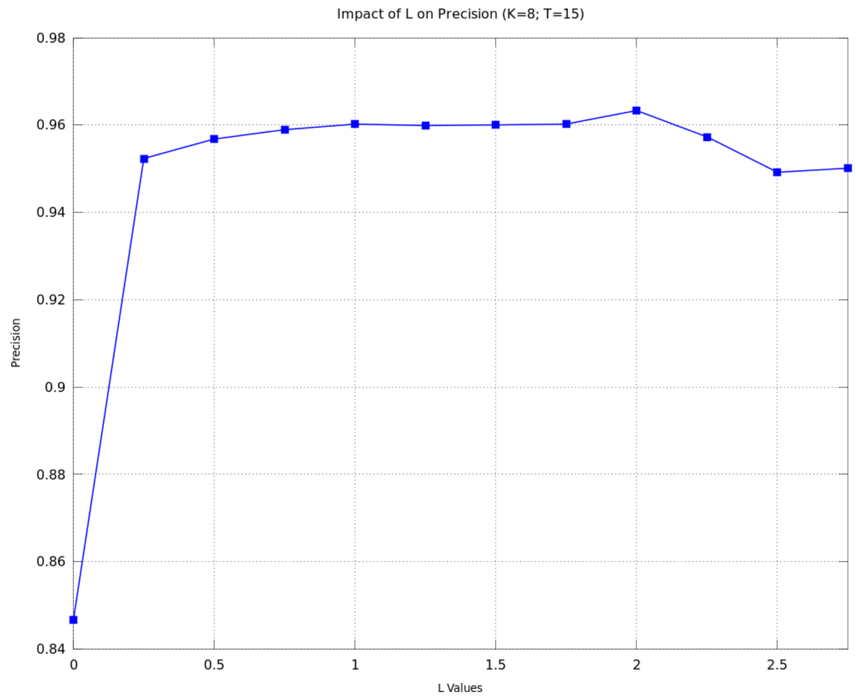

Figure 6: Impact of parameter L ($K = 8$, $T = 15$).

or shape). Methods proposed in [13, 22, 42, 43] used shape descriptors, while the method proposed in [20] used a color descriptor. In [21], an evaluation involving shape, color, and texture was presented. Our goal here is to evaluate the use of our method for several CBIR tasks involving shape, color, and texture descriptors. The measure adopted is *Mean Average Precision (MAP)*, geometrically referred to as the mean area below precision $\times$ recall curve. Next subsections describe descriptors and datasets used for shape, color, and texture experiments. Results are presented in Table 3. As we can observe, the Pairwise Recommendation Re-Ranking method presents positive effectiveness gains for all descriptors (including shape, color, and texture), ranging from +1.27% to +20.47%. We conducted a paired t-test and conclude that there is a 99%

Table 2: Pairwise Recommendation for shape descriptors on the MPEG-7 dataset *(Recall@40)*.

Shape Descriptor	**Score**	**Pairwise Recommendation**	**Gain**
SS [30]	43.99%	54.36%	+23.57%
BAS [1]	75.20%	84.03%	+11.74%
IDSC [16]	85.40%	92.21%	+7.97%
CFD [22]	84.43%	96.15%	+13.88%

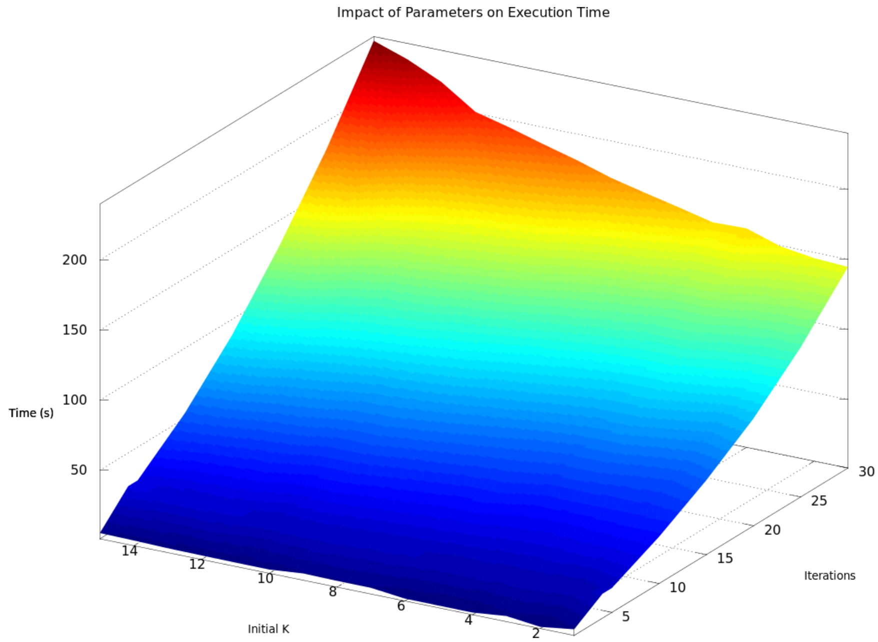

Figure 7: Impact of parameters on execution time.

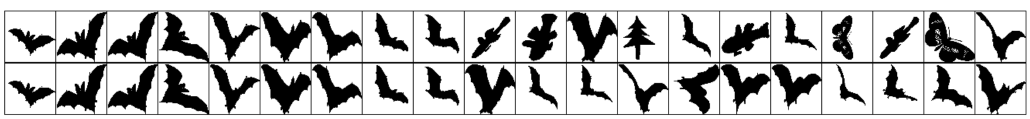

Figure 8: First row: retrieval results for the CFD [22] shape descriptor (first image is taken as the query image). Second row: retrieval results for the same shape descriptor after the use of the Pairwise Recommendation algorithm.

chance of difference between the means (before and after the re-ranking) being statistical significantly.

4.3.1. Shape Descriptors

For the experiments with the shape collection, we used the same descriptors and dataset considered in the previous section, but now using MAP as effectiveness measure. Results are similar to those obtained considering the bulleyes score, with positive gains ranging from +5.92% to +13.22%.

4.3.2. Texture Descriptors

In this section, we aim at validating our method using texture descriptors. The experiments consider three well-known texture descriptors: Local Binary Patterns (LBP) [18], Color Co-Occurrence Matrix (CCOM) [14], and Local Activity Spectrum (LAS) [36]. We used the Brodatz [5] dataset, a popular dataset for texture descriptor evaluation. The Brodatz dataset is composed of

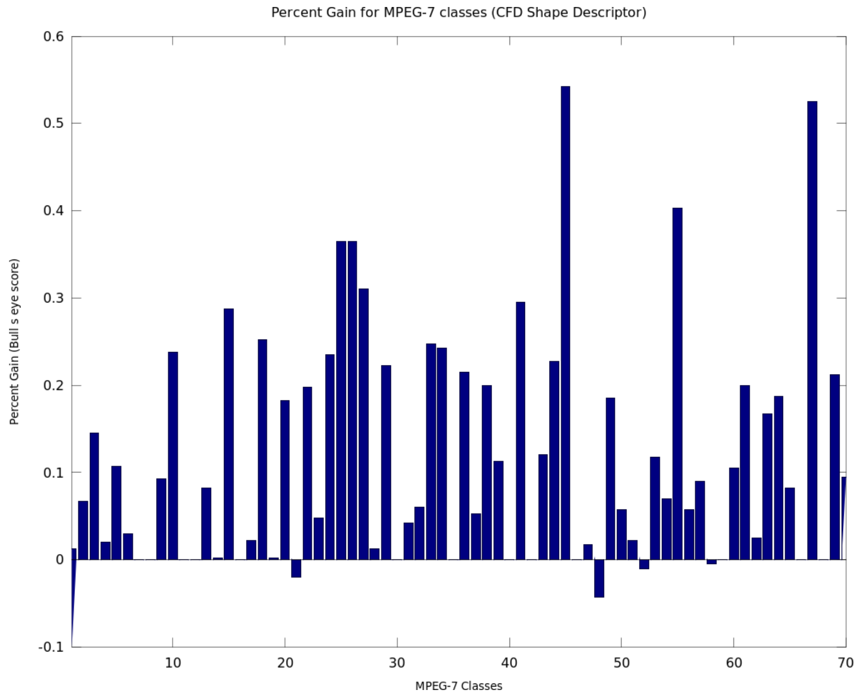

Figure 9: Percentage gains in bulls-eye score for each class of the MPEG-7 dataset considering the CFD [22] shape descriptor.

111 different textures of size 512 $\times$ 512 pixels. Each texture is divided into 16 blocks 128 $\times$ 128 pixels of non-overlapping subimages, such that 1776 images are considered. Some examples of textures are illustrated in Figure 10. We use the same values for the parameters of the re-ranking method, except for L. In this experiment, we set $L = 1$. Our re-ranking method presents positive gains ranging from +7.27% to 15.44%.

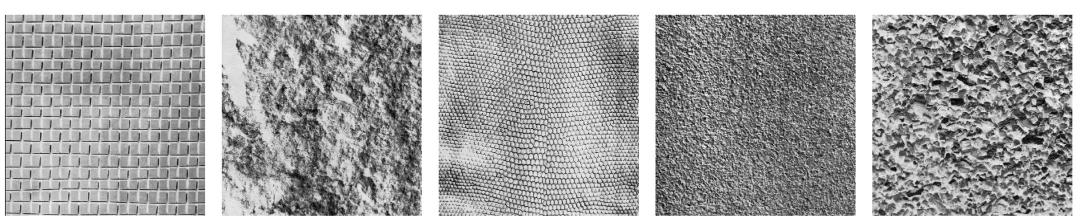

Figure 10: Examples of Brodatz [5] texture images.

4.3.3. Color Descriptors

We evaluate our method for three color descriptors: Border/Interior Pixel Classification (BIC) [34], Auto Color Correlograms (ACC) [11], and Global

Table 3: Pairwise Recommendation evaluation on several content-based image retrieval tasks *(MAP)*.

Image Descriptor	**Type**	**Dataset**	**Score** *(MAP)*	**Pairwise Recomm.**	**Gain**
SS [30]	Shape	MPEG-7	37.67%	39.90%	+5.92%
BAS [1]	Shape	MPEG-7	71.52%	77.65%	+8.57%
IDSC [16]	Shape	MPEG-7	81.70%	86.83%	+6.28%
CFD [22]	Shape	MPEG-7	80.71%	91.38%	+13.22%
GCH [35]	Color	Soccer	32.24%	32.35%	+0.34%
ACC [11]	Color	Soccer	37.23%	40.31%	+8.27%
BIC [34]	Color	Soccer	39.26%	42.64%	+8.61%
LBP [18]	Texture	Brodatz	48.40%	51.92%	+7.27%
CCOM [14]	Texture	Brodatz	57.57%	66.46%	+15.44%
LAS [36]	Texture	Brodatz	75.15%	80.73%	+7.43%

Color Histogram (GCH) [35]. The experiments were conducted on a database used in [40]. The Soccer dataset is composed by 280 images from 7 soccer teams, containing 40 images per class. The size of images range from 198 $\times$ 148 to 537 $\times$ 672 pixels. Some samples of this dataset are shown in Figure 11. We can observe a positive gain for all color descriptors ranging from 0.34% to 8.61% (for the MAP effectiveness measure).

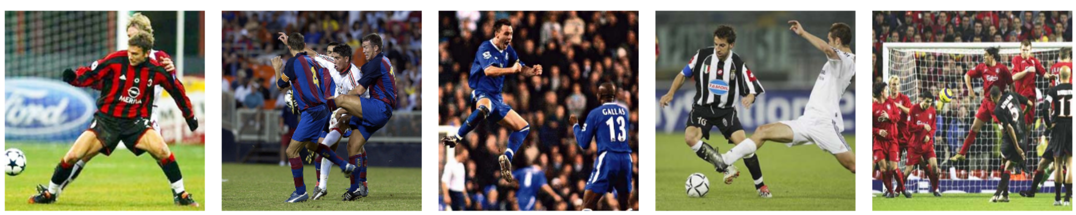

Figure 11: Examples of Soccer dataset [40] images.

4.4. Analysis of Convergence

This section aims at discussing and experimentally evaluating the convergence of the proposed re-ranking method.

Figure 12 shows the evolution of cohesion measure, whose variation is used as convergence criterion. We considered three different descriptors/datasets: the CFD [22] shape descriptor on the MPEG-7 dataset, the BIC [34] color descripor on the Soccer dataset, and LAS [36] texture descriptor on the Brodatz dataset. We can observe a similar behavior for the three curves: at the beginning, the cohesion measure increases quickly and, at the end, it converges for a constant value. As discussed in Section 3.6, in scenarios with less effective descriptors, the convergence is slower. That can be observed for the BIC [34] descriptor on the Soccer dataset, which has the lowest effectiveness performance.

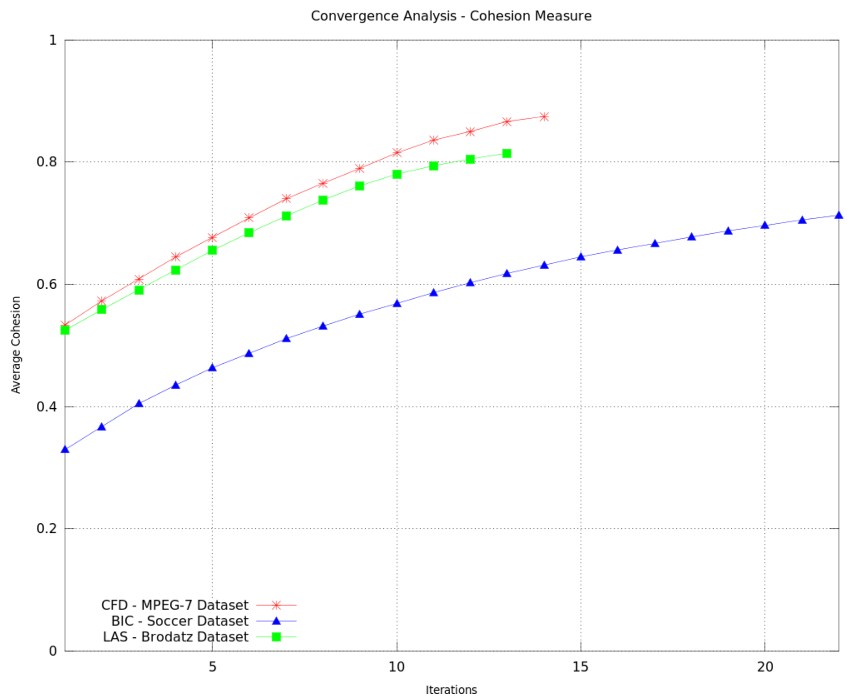

Figure 12: Analysis of convergence - Cohesion measure.

Besides the cohesion measure, we also consider the difference between ranked lists along iterations. Intuitively, we consider that an iterative re-ranking algorithm converges if, after a certain number of iterations, it produces a small number of changes in the generated ranked lists. More formally, we consider a definition of ε-convergence for rankings presented in [26]:

Let $\mathcal{C}=\{img_1, img_2, \ldots, img_N\}$ be an *image collection* and let R be a *ranked list* $R=\{img_1,\ img_2,\ \ldots,\ img_N\}$, which can be defined as a permutation of the collection $\mathcal{C}$. An iterative ranking algorithm that generates a ranked list $R(t)$ at each iteration t, ε-*converges* in (at most) T iterations in a metric $d(\cdot,\cdot)$, if there exists a ranked list R such that, for every $t \geq T$, $d(R(t), R) < \varepsilon$.

A natural distance function to use for this definition is the Kendall's tau metric. This metric is equal to the number of exchanges needed in a bubble sort to convert one permutation to the other. The use of this metric for comparing top-k lists is presented in [8].

In this scenario, we have measured the evolution of Kendall's tau distance between rankings at each iteration for the three descriptors. For measuring the Kendall's tau distance, we considered the $2 \times K$ top images of ranked lists (same size considered for cohesion measure). Figure 13 shows the evolution of average Kendall's tau distance between rankings along iterations.

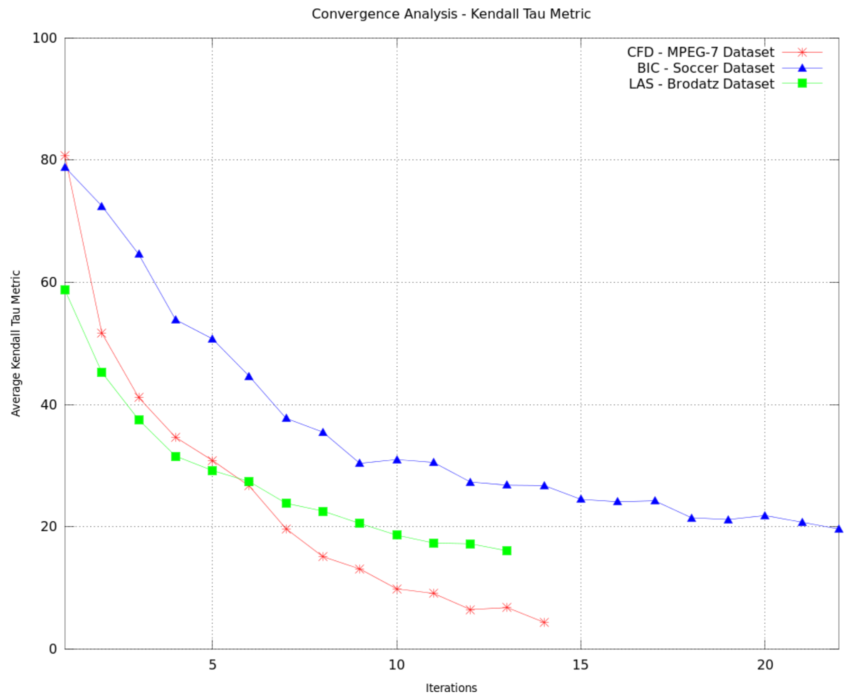

Figure 13: Analysis of convergence - Kendall's tau distance.

The results obtained by using the Kendall's tau distance is consistent with the cohesion measure evolution (Figure 12). The Kendall's tau distance *decreases* at the same pace as the cohesion measure *increases*. A high distance can be observed at first iterations, indicating a lot of changes in the ranked lists. After some iterations, the convergence criterion is reached (distances get lower values).

The same results were observed for the other descriptors used in the three image collections. On average, all descriptors converged in 17 iterations.

4.5. Post-Processing Methods

We also evaluated our method in comparison with other state-of-the-art post-processing methods. We consider other five post-processing methods (used with various shape descriptors) and two different datasets: MPEG-7 and Kimia99. In Table 4, we present the results for MPEG-7 dataset. The forth column shows the bullseyes scores for shape descriptors. The fifth column presents the scores for the combination of shape descriptor and post-processing method. Since we are among the few in the literature to consider statistical significance, we could not obtain the dispersion measures for published results. Therefore, we have computed the confidence intervals of our results (for re-ranking and rank aggregation tasks, detailed in next section) and considered ourselves tied to

Table 4: Post-processing methods comparison on the MPEG-7 dataset *(Recall@40)*.

Algorithm	**Descriptor**	**Score [%]**	**Gain**
Contour Features Descriptor (CFD) [22]	-	84.43%	-
Inner Distance Shape Context (IDSC) [16]	-	85.40%	-
Aspect Shape Context (ASC) [17]	-	88.39%	-
Graph Transduction (LP) [42]	IDSC	91.00%	+6.56%
Distance Optimization [22]	CFD	92.56%	+9.63%
Locally Constrained Diffusion Process (LCDP) [43]	IDSC	93.32%	+9.27%
Mutual kNN Graph [13]	IDSC	93.40%	+9.37%
Contextual Re-Ranking [21]	CFD	94.55%	+11.99%
Locally Constrained Diffusion Process (LCDP) [43]	ASC	95.96%	+8.56%
Pairwise Recommendation	**CFD**	**96.15%**	**+13.88%**

Table 5: Post-processing methods comparison on the Kimia99 dataset.

Algorithm	**Descriptor**	**1º**	**2º**	**3º**	**4º**	**5º**	**6º**	**7º**	**8º**	**9º**	**10º**
CFD [22]	-	99	98	98	99	97	90	86	86	68	56
IDSC [16]	-	99	99	99	98	98	97	97	98	94	79
Distance Optimization [22]	CFD	98	99	99	99	98	99	99	97	98	99
Graph Transduction [42]	IDSC	99	99	99	99	99	99	99	99	97	99
Mutual kNN Graph [13]	IDSC	99	99	99	99	99	99	99	99	99	99
Pairwise Recommendation	**CFD**	**99**	**99**	**99**	**99**	**99**	**99**	**99**	**99**	**99**	**99**

those published results which are in our interval. We obtained 96.15% (95% confidence interval: [95.22%, 97.09%]).

Note that the results of our method (in bold) present the best effectiveness performance when compared to all other post-processing methods. Table 5 presents the comparison on Kimia99 dataset. Scores are calculated as the sum of correctly retrieved shapes from all classes within the first 10 objects. Therefore, the best resulting score for each of them is 99. Note that the maximum retrieval score was reached by our method (in bold).

4.6. Rank Aggregation

This section aims at evaluating the use of our re-ranking method for combining different CBIR descriptors. We selected two descriptors considering each visual property. Descriptors with best effectiveness results were selected. Table 6 presents the results of MAP score for these descriptors. We observe significant gains compared with each the use of each descriptor in isolation. Figure 14 shows the Precision $\times$ Recall curves of shape descriptors CFD [22] and IDSC [16] in different situations: before and after using the Pairwise Recommendation Re-Ranking algorithm, and after their combination using our re-ranking

Table 6: Pairwise Recommendation for descriptor combination *(MAP)*

Descriptor	**Type**	**Dataset**	**Score**[%]
CFD [22] + IDSC [16]	Shape	MPEG-7	98.78%
BIC [34] + ACC [11]	Color	Soccer	42.20%
LAS [36] + CCOM [14]	Texture	Brodatz	79.91%

method. We also compared our method with other combination approaches on the MPEG-7 dataset. Results are presented in Table 7. We can observe that the Pairwise Recommendation method achieves the best bulleyes score. For rank aggregation tasks, it yields 99.52% (95% confidence interval: [99.22%, 99.82%]).

Finally, we analyze the impact of our combination method on the distance matrix. Figure 15 illustrates a subset (200 x 200) of distance matrices for the MPEG-7 dataset considering descriptors CFD [22], IDSC [16] and the combination using the Pairwise Recommendation algorithm. The dark pixels indicate low distances between images. In matrix which represents the combination using the Pairwise Recommendation algorithm, we can observe very distinct squares that illustrate the low distances among shapes from the same classes.

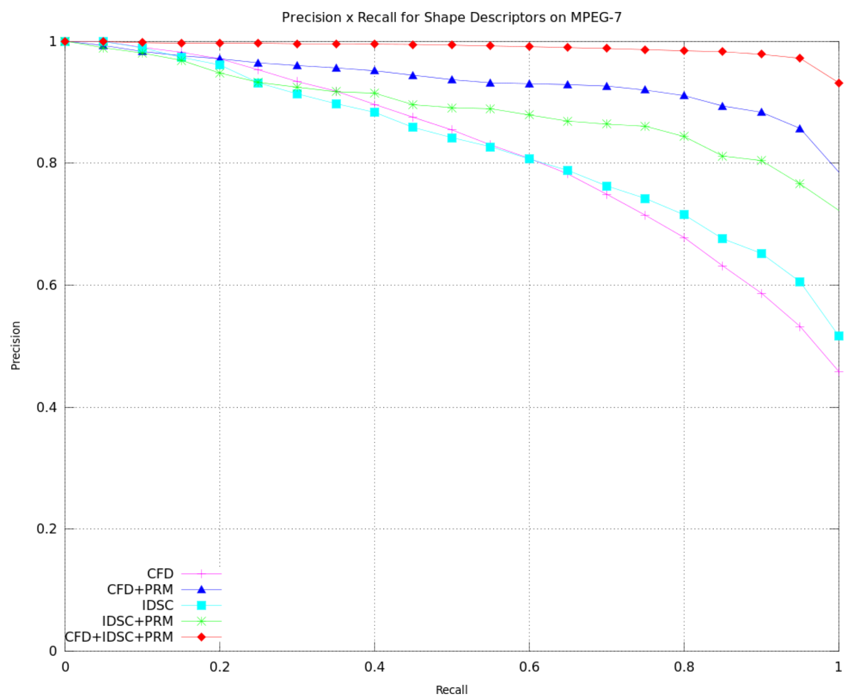

Figure 14: Pairwise Recommendation for shape descriptors on the MPEG-7 dataset.

Table 7: Comparison of Combination Aproaches on MPEG-7 database *(Recall@40)*.

Algorithm	**Score [%]**
IDSC [16]+Strategy II [37] + LCDP [43]	93.80%
IDSC [16]+Strategy I [37] + LCDP [43]	94.85%
IDSC [16]+Strategies I&II [37] + LCDP [43]	95.60%
IDSC [16]+DDGM [39]+Co-Transduction [2]	97.31%
SC [3]+DDGM [39]+Co-Transduction [2]	97.45%
SC [3]+IDSC [16]+Co-Transduction [2]	97.72%
CFD [22]+IDSC [16]+Pairwise Recommendation	**99.52%**

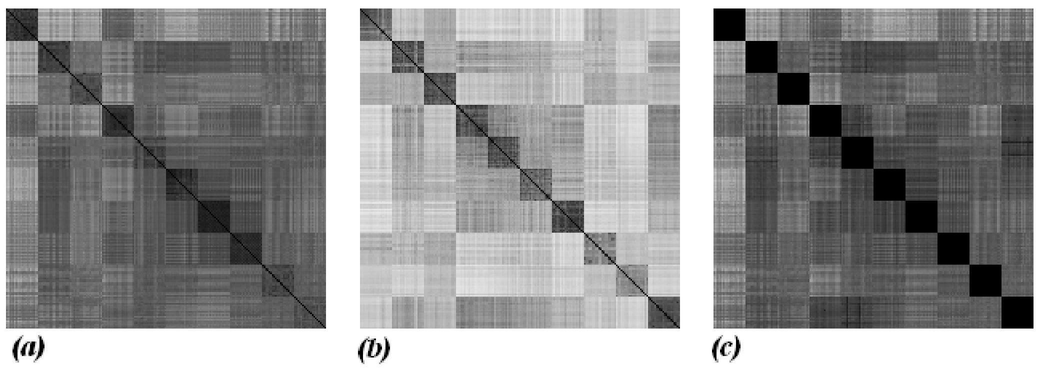

Figure 15: Distance matrices by shape descriptors on the MPEG-7 dataset: (a) CFD [22]; (b) IDSC [16]; (c) CFD+IDSC+Pairwise Recommendation.

5. Conclusions

In this paper, we presented a novel re-ranking approach based on the concept of recommendation. The main idea consists in using information encoded in ranked lists for making recommendations (updating distances) among images. We conducted a large set of experiments and experimental results demonstrate the effectiveness of our method in several image retrieval tasks based on shape, color, and texture descriptors. The proposed method achieves very high effectiveness performance when compared with state-of-the-art post-processing methods on the well-known datasets.

In future work, we intend to investigate the use of different collaborative filtering techniques with our method. The combination of our method with techniques that exploit user interactions (as relevance feedback, for example) are very promising and are also left for future work.

6. Acknowledgments

Authors thank AMD, CAPES, FAPESP, FAEPEX, and CNPq for financial support. Authors also thank DGA/UNICAMP for its support in this work.

References

[1] Arica, N., Vural, F.T.Y.. Bas: a perceptual shape descriptor based on the beam angle statistics. Pattern Recognition Letters 2003;24(9-10):1627–1639.

[2] Bai, X., Wang, B., Wang, X., Liu, W., Tu, Z.. Co-transduction for shape retrieval. In: European Conference on Computer Vision. volume 3; 2010. p. 328–341.

[3] Belongie, S., Malik, J., Puzicha, J.. Shape matching and object recognition using shape contexts. IEEE Transactions on Pattern Analysis and Machine Intelligence 2002;24(4):509–522.

[4] Blanco-Fernández, Y., López-Nores, M., Gil-Solla, A., Ramos-Cabrer, M., Pazos-Arias, J.J.. Exploring synergies between content-based filtering and Spreading Activation techniques in knowledge-based recommender systems. Information Sciences 2011;181(21):4823–4846.

[5] Brodatz, P.. Textures: A Photographic Album for Artists and Designers. Dover, 1966.

[6] Ciocca, G., Cusano, C., Santini, S., Schettini, R.. Halfway through the semantic gap: Prosemantic features for image retrieval. Information Sciences 2011;181(22):4943–4958.

[7] Diaz, F.. Regularizing ad hoc retrieval scores. In: ACM Conference on Information and Knowledge Management. 2005. p. 672–679.

[8] Fagin, R., Kumar, R., Sivakumar, D.. Comparing top k lists. In: ACM-SIAM Symposium on Discrete Algorithms. 2003. p. 28–36.

[9] Faria, F.F., Veloso, A., Almeida, H.M., Valle, E., da S. Torres, R., Goncalves, M.A., Jr., W.M.. Learning to rank for content-based image retrieval. In: Multimedia Information Retrieval. 2010. p. 285–294.

[10] Gonçalves, M.A.. Streams, Structures, Spaces, Scenarios, and Societies (5S): A Formal Digital Library Framework and Its Applications. Ph.D. thesis; Virginia Polytechnic Institute and State University; 2004.

[11] Huang, J., Kumar, S.R., Mitra, M., Zhu, W.J., Zabih, R.. Image indexing using color correlograms. In: IEEE Conference on Computer Vision and Pattern Recognition. 1997. p. 762.

[12] Ji, S., Zhou, K., Liao, C., Zheng, Z., Xue, G.R., Chapelle, O., Sun, G., Zha, H.. Global ranking by exploiting user clicks. In: ACM SIGIR Conference on Research and Development in Information Retrieval. 2009. p. 35–42.

[13] Kontschieder Peter Donoser Michael, B.H.. Beyond pairwise shape similarity analysis. In: Asian Conference on Computer Vision. 2009. p. 655–666.

[14] Kovalev, V., Volmer, S.. Color co-occurence descriptors for querying-by-example. In: International Conference on Multimedia Modeling. 1998. p. 32.

[15] Latecki, L.J., Lakmper, R., Eckhardt, U.. Shape descriptors for non-rigid shapes with a single closed contour. In: IEEE Conference on Computer Vision and Pattern Recognition. 2000. p. 424–429.

[16] Ling, H., Jacobs, D.W.. Shape classification using the inner-distance. IEEE Transactions on Pattern Analysis and Machine Intelligence 2007;29(2):286–299.

[17] Ling, H., Yang, X., Latecki, L.J.. Balancing deformability and discriminability for shape matching. In: European Conference on Computer Vision. volume 3; 2010. p. 411–424.

[18] Ojala, T., Pietikäinen, M., Mäenpää, T.. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Transactions on Pattern Analysis and Machine Intelligence 2002;24(7):971–987.

[19] Page, L., Brin, S., Motwani, R., Winograd, T.. The PageRank Citation Ranking: Bringing Order to the Web. Technical Report 1999-66; Stanford InfoLab; 1999.

[20] Park, G., Baek, Y., Lee, H.K.. Re-ranking algorithm using post-retrieval clustering for content-based image retrieval. Information Processing and Management 2005;41(2):177 – 194.

[21] Pedronette, D.C.G., da S. Torres, R.. Exploiting contextual information for image re-ranking. In: Iberoamerican Congress on Pattern Recognition. 2010. p. 541–548.

[22] Pedronette, D.C.G., da S. Torres, R.. Shape retrieval using contour features and distance optmization. In: International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications. volume 1; 2010. p. 197 – 202.

[23] Pedronette, D.C.G., da S. Torres, R.. Exploiting clustering approaches for image re-ranking. Journal of Visual Languages and Computing 2011;22(6):453–466.

[24] Pedronette, D.C.G., da S. Torres, R.. Image re-ranking and rank aggregation based on similarity of ranked lists. In: Computer Analysis of Images and Patterns. volume 6854; 2011. p. 369–376.

[25] Perronnin, F., Liu, Y., Renders, J.M.. A family of contextual measures of similarity between distributions with application to image retrieval. In: IEEE Conference on Computer Vision and Pattern Recognition. 2009. p. 2358–2365.

[26] Peserico, , E. Pretto, L.. What does it mean to converge in rank? In: Proceedings of 1st International Conference on the Theory of Information. 2007. p. 239–245.

[27] Qin, T., Liu, T.Y., Zhang, X.D., Wang, D.S., Li, H.. Global ranking using continuous conditional random fields. In: Neural Information Processing Systems. 2008. p. 1281–1288.

[28] van Rijsbergen, C.J.. Information Retrieval. 2nd ed., 1979.

[29] da S. Torres, R., ao, A.X.F.. Content-Based Image Retrieval: Theory and Applications. Revista de Informática Teórica e Aplicada 2006;13(2):161–185.

[30] da S. Torres, R., Falcao, A.X.. Contour Salience Descriptors for Effective Image Retrieval and Analysis. Image and Vision Computing 2007;25(1):3–13.

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

[32] Schwander, O., Nielsen, F.. Reranking with contextual dissimilarity measures from representational bregmanl k-means. In: International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications. volume 1; 2010. p. 118–122.

[33] Serrano-Guerrero, J., Herrera-Viedma, E., Olivas, J.A., Cerezo, A., Romero, F.P.. A google wave-based fuzzy recommender system to disseminate information in university digital libraries 2.0. Information Sciences 2011;181:1503–1516.

[34] Stehling, R.O., Nascimento, M.A., Falcao, A.X.. A compact and efficient image retrieval approach based on border/interior pixel classification. In: ACM Conference on Information and Knowledge Management. 2002. p. 102–109.

[35] Swain, M.J., Ballard, D.H.. Color indexing. International Journal of Computer Vision 1991;7(1):11–32.

[36] Tao, B., Dickinson, B.W.. Texture recognition and image retrieval using gradient indexing. Journal of Visual Communication and Image Representation 2000;11(3):327 – 342.

[37] Temlyakov, A., Munsell, B.C., Waggoner, J.W.,, , Wang, S.. Two perceptually motivated strategies for shape classification. In: IEEE Conference on Computer Vision and Pattern Recognition. volume 1; 2010. p. 2289–2296.

[38] Thang, N.D., Rasheed, T., Lee, Y.K., Lee, S., Kim, T.S.. Content-based facial image retrieval using constrained independent component analysis. Information Sciences 2011;181(15):3162–3174.

[39] Tu, Z., Yuille, A.L.. Shape matching and recognition - using generative models and informative features. In: European Conference on Computer Vision. 2004. p. 195–209.

[40] van de Weijer, J., Schmid, C.. Coloring local feature extraction. In: European Conference on Computer Vision. volume Part II; 2006. p. 334–348.

[41] Yang, L., Ji, D., Zhou, G., Nie, Y., Xiao, G.. Document re-ranking using cluster validation and label propagation. In: ACM Conference on Information and Knowledge Management. 2006. p. 690–697.

[42] Yang, X., Bai, X., Latecki, L.J., Tu, Z.. Improving shape retrieval by learning graph transduction. In: European Conference on Computer Vision. volume 4; 2008. p. 788–801.

[43] Yang, X., Koknar-Tezel, S., Latecki, L.J.. Locally constrained diffusion process on locally densified distance spaces with applications to shape retrieval. In: IEEE Conference on Computer Vision and Pattern Recognition. 2009. p. 357–364.

[44] Zhu, X.. Semi-supervised learning with graphs. Ph.D. thesis; Pittsburgh, PA, USA; 2005. Chair-Lafferty, John and Chair-Rosenfeld, Ronald.