Preprint version of paper published in Information Sciences journal, 2012.
DOI: http://doi.org/10.1016/j.ins.2013.12.030

# A Scalable Re-Ranking Method for Content-Based Image Retrieval

Daniel Carlos Guimarães Pedronette[1], Jurandy Almeida[2], and Ricardo da S. Torres[3]

[1] *Department of Statistics, Applied Mathematics and Computing (DEMAC), State University of São Paulo (UNESP), Rio Claro - Brazil*
[2] *Institute of Science and Technology, Federal University of São Paulo (UNIFESP), São Jose dos Campo - Brazil*
[3] *RECOD Lab, Institute of Computing (IC), University of Campinas (UNICAMP), Campinas - Brazil*
*daniel@rc.unesp.br, jurandy.almeida@unifesp.br , rtorres@ic.unicamp.br*

**Abstract**

Content-based Image Retrieval (CBIR) systems consider only a pairwise analysis, *i.e.*, they measure the similarity between pairs of images, ignoring the rich information encoded in the relations among several images. However, the user perception usually considers the query specification and responses in a given *context*. In this scenario, re-ranking methods have been proposed to exploit the *contextual information* and, hence, improve the effectiveness of CBIR systems. Besides the *effectiveness*, the usefulness of those systems in real-world applications also depends on the *efficiency* and *scalability* of the retrieval process, imposing a great challenge to the re-ranking approaches, once they usually require the computation of distances among all the images of a given collection. In this paper, we present a novel approach for the re-ranking problem. It relies on the similarity of top-$k$ lists produced by efficient indexing structures, instead of using distance information from the entire collection. Extensive experiments were conducted on a large image collection, using several indexing structures. Results from a rigorous experimental protocol show that the proposed method can obtain significant effectiveness gains (up to 12.19% better) and, at the same time, improve considerably the efficiency (up to 73.11% faster). In addition, our technique scales up very well, which makes it suitable for large collections.

*Keywords:* content-based image retrieval; re-ranking methods; indexing structures

## 1. Introduction

Advances in multimedia technologies for creating and sharing digital contents have triggered an exponential increase of image collections. In order to deal with these collections, it is necessary to develop methods for efficiently indexing and retrieving these data. Traditional search approaches based on image metadata and keywords can be unfeasible for large collections, since manual annotation is prohibitively expensive. In this scenario, Content-Based Image Retrieval (CBIR) systems [10, 35] have emerged as an alternative to overcome those limitations by taking into account the content of the images for supporting retrieval tasks.

A common task for CBIR systems is to retrieve the most similar images to a query pattern (*e.g.*, query image) defined by users. In general, the output provided is a ranked list, where the images are disposed in decreasing order of similarity, according to a visual property, such as shape, color, and texture. In this scenario, accurately ranking the collection images is of great relevance. Existing systems often consider only pairwise analysis, measuring the similarity between pairs of images and ignoring the relevant information encoded in the relations among several images. The user perception, on the other hand, considers the query specification and responses in a given *context*.

Motivated by these limitations, many supervised learning approaches have been proposed. Relevance Feedback [13, 37, 34, 41, 53] and Active Re-Ranking [42] methods, for example, were incorporated into CBIR systems with the aim of exploiting interactions for learning users needs. Basically, the image retrieval process with relevance feedback is comprised of four steps: (i) showing a small number of retrieved images to the user; (ii) user indication of relevant and non-relevant images; (iii) learning the user needs by taking into account his/her feedbacks; (iv) and selecting a new set of images to be shown. This procedure is repeated until a satisfactory result is reached. Although very effective, these approaches require a lot of human efforts for obtaining enough training data, which can be infeasible for some real-world systems.

Aiming at overcoming these problems, efforts were put on unsupervised approaches. Recently, various approaches [19, 29, 49, 50] have been proposed to improve the *effectiveness* of retrieval tasks by taking into account the relationships among all dataset objects. In other words, research efforts have been focused on post-processing the similarity (or distance) scores, by using the contextual information available in relationships among images of a given collection. The goal of those methods is somehow mimic the human behavior on judging the similarity among objects by considering specific contexts.

The key advantage of re-ranking approaches consists in the fact that they require no user intervention, training or labeled data, operating on an absolutely unsupervised way.

However, the usefulness of re-ranking approaches for CBIR systems depends not only on the *effectiveness*, but also on the *efficiency* and *scalability*. While the effectiveness is related to the quality of retrieved images, the efficiency refers to the time spent to obtain the results. Scalability considers the system capability of handling growing image collections. Although the effectiveness has been the focus of various recently works [27, 46, 49], dealing with those three requirements at the same time is essential in real-world applications. Aiming at computing the relationship among images, re-ranking algorithms often consider all the distances among images of a given dataset, which represent a large computational effort (typically, between $O(N^2)$ and $O(N^3)$), hindering its use in searching services that deal with real-world image collections.

On the other hand, significant research efforts have been spent trying to improve the performance in processing similarity queries. Most of existing indexes employed to accelerate data retrieval are constructed by partitioning a set of objects using distance-based criteria. Those approaches avoid the computation of distances among all the images of a given collection.

In this paper, we aim at combining the potential of effectiveness gains obtained by the re-ranking approaches with the power of the indexing structures in processing similarity queries efficiently. Here, we present a novel approach for the re-ranking problem that relies on ranked lists produced by efficient indexing structures. The ranked lists used by the proposed method contain only a subset of the most similar images, avoiding the computation, storage, and processing of distance information from the entire collection.

The main contribution of the proposed index-based re-ranking approach consists in its capacity of combining effectiveness and efficiency features, making it suitable for large collections. The proposed re-ranking method requires very low computational efforts, presenting an asymptotic complexity of only $O(N)$. On the other hand, the effectiveness gains are comparable to state-of-the-art approaches.

We carried out extensive experiments on a large image collection, considering several indexing structures. The reported results demonstrate that the proposed method obtains significant effectiveness gains (up to 12.19% better) and, at the same time, improves considerably the efficiency (up to 73.11% faster). Moreover, our technique scales up very well, which makes it suitable for large collections. We also evaluated the proposed method in comparison with several other state-of-the-art approaches considering a common shape dataset. Experimental results demonstrate that the proposed method yields effectiveness results comparable to post-processing algorithms recently proposed in the literature.

The remainder of this paper is organized as follows. Section 2 describes related work. Section 3 introduces the re-ranking problem based on ranked lists. Section 4 presents our image re-ranking approach. Section 5 discusses indexing structures used to produce ranked lists. Section 6 reports the results of our experiments. Finally, we offer our conclusions and directions for future work in Section 7.

## 2. Related Work

This section presents related work. Section 2.1 overviews image re-ranking approaches, while existing indexes structures are discussed in Section 2.2.

### *2.1. Image Re-Ranking*

In recent years, several CBIR approaches [19, 27, 29, 46, 49, 51] have been proposed aiming at improving the effectiveness of retrieval tasks by replacing pairwise similarities by more global affinities that also consider the relation among the database objects [51].

Although using a very diverse taxonomy (re-ranking [24, 29], graph transduction [49], diffusion process [50], affinity learning [51], contextual similarity/dissimilarity measures [30, 46]), these *post-processing* methods have in common the goal of improving the effectiveness of retrieval tasks by exploiting considering relationships among dataset objects on an unsupervised way, requiring no training data.

Graph-based methods are used by several approaches [19, 46, 49]. In [49], a *graph-based transductive learning algorithm* is proposed for shape retrieval tasks. It learns a better metric through a graph transduction by propagating the model through existing shapes, in a similar manner to the computation of geodesics in a dataset manifold. Another approach based on propagating the similarity information in a weighted graph is proposed in [51] as *affinity learning*. Instead of propagating the similarity information in the original graph, it uses a tensor product graph (TPG) obtained by the tensor product of the original graph with itself. A shortest path propagation algorithm is proposed in [46], which is a graph-based algorithm for shape/object retrieval.

Given a query object and a target database object, it explicitly finds the shortest path between them in the distance manifold of the database objects.

Another strategy frequently used consists in employing clustering methods [24, 38]. A re-ranking framework for CBIR systems based on contextual dissimilarity measures is proposed in [38]. The contexts are modeled using a clustering algorithm to group similar images from the ranked list. In [24], a re-ranking algorithm using post-retrieval clustering for CBIR is proposed. In the first step, images are retrieved using visual features, such as color histograms. Next, the retrieved images are analyzed using hierarchical agglomerative clustering methods and the rank of the results is adjusted according to the distance of a cluster to a query.

Aiming at computing the relationship among images, re-ranking algorithms commonly consider all the distances among images of a given dataset. In general, a distance matrix is the input of those algorithms, which represents a large computational effort, assuming a complexity $O(N^2)$, where $N$ is the dataset size. Other approaches consider complete ranked lists, which contain distance information of the entire collection ordered according to their similarity to the query image. The processing time in this case is $O(N)$.

However, the most important information is found in the top positions of ranked lists, which are expected to contain the most similar images to the query image. Therefore, it can be very valuable an strategy that considers only a subset of the ranked lists, with size less than $N$. It is valid specially for large collections, where $N$ is very high, and therefore the ranked lists are very expensive to compute.

In this scenario, this paper presents two important contributions: $(i)$ the introduction of an image re-ranking method that does not require the computation of distances among all the images or complete ranked lists; and $(ii)$ the use of efficient indexing structures for obtaining the ranked lists.

### *2.2. Indexing Structures*

Traditional database systems [11, 31] are able to efficiently deal with structured records by using the *exact match* paradigm. However, complex data types, such as multimedia data (audio, image, and video), biological data (genomic and protein sequences), among others, cannot be represented effectively as structured records [55].

In those cases, *similarity search* [18] has been established as a fundamental paradigm. Essentially, the problem is to find, in a set of objects, those which are the most similar to a given query object. The similarity between any pair of objects is computed by some distance function, being understood that low values of distance correspond to high degrees of similarity [55].

The commonest types of similarity queries include (1) *range* queries, where all the objects whose distance to the query does not exceed a threshold are requested; and (2) *k-nearest neighbors* (kNN) queries, where a specified number $k$ of objects, which are closest to the query are requested [55].

Several index structures have been proposed to speed up similarity queries [3, 4, 8, 14, 33]. They can be broadly classified, depending on their field of applicability, as *multi-dimensional* (or spatial) and *metric access methods*, where the use of the former is only possible when the feature space is a vector space [55].

Algorithms to search in general metric spaces can be divided into two large areas: pivot-based and clustering-based methods [8]. A pivot-based strategy selects some objects as *pivots* from the collection and then computes and stores the distances between the pivots and the objects of the database. During the search, those distances are used to discard objects without comparing them with the query. Clustering techniques consist in dividing the space into zones as compact as possible, normally in a recursive fashion, and storing a *representative* ("center") for each zone plus a few extra data that allows us to quickly discard the zone at query time. In a search, complete regions are discarded by using the distances from their representatives to the query [8].

Two criteria can be used to delimit a zone in the clustering-based approaches. The first one selects a set of representatives and put each other object inside the zone of its closest representative, thus the areas are limited by *hyperplanes*. The second criterion is the *covering radius*, which is the maximum distance between a representative and any object in its zone [8].

## 3. Problem Definition

Let $\mathcal{C}=\{img_1, img_2, \ldots, img_N\}$ be an *image collection*, where $N$ is the cardinality $|\mathcal{C}|$ of collection $\mathcal{C}$. Let $\mathcal{D}$ be an *image descriptor* which can be defined [36] as a tuple $(\epsilon, \rho)$, where:

- $\epsilon$: $\hat{I} \rightarrow \mathbb{R}^n$ *is a function, which extracts a feature vector* $v_{\hat{I}}$ *from an image* $\hat{I}$*.*
- $\rho$: $\mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$ *is a distance function that computes the distance between two images by means of 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 $\rho(img_i,img_j)$ for the denoting the distance between images $img_i$ and $img_j$.

Based on the distance function $\rho$, given a query image $img_q$, we can compute a ranked list $\sigma_q$ in response to the query. The ranked lists can contain distance information from the entire collection, sorted according to their similarity to the query image. However, the top positions of ranked lists are expected to contain the most relevant images related to the query image. Therefore, it can be very desirable that the ranked list $\sigma_q$ considers only a subset of the $N_S$ most similar images, where $N_S < N$ is the number of images at top positions of the ranked list that we would like to consider. It is valid specially for large collections, where $N$ is very high, and therefore $\sigma_q$ is very expensive to compute.

In this way, the ranked list $\sigma_q$=$(img_1,\ img_2,\ \ldots,\ img_{N_S})$ can be defined as a permutation of the image collection $\mathcal{C}_S \subset \mathcal{C}$, which contains the most similar images to query image $img_q$, such that and $|\mathcal{C}_S| = N_S$. A permutation $\sigma_q$ is as a bijection from the collection $\mathcal{C}_S$ onto the set $[N_S] = \{1, 2, \ldots, N_S\}$. For a permutation $\sigma_q$, we interpret $\sigma_q(i)$ as the position (or rank) of image $img_i$ in the ranked list $\sigma_q$. Therefore, we can say that, if $img_x$ is ranked before $img_y$, that is $\sigma_q(x) < \sigma_q(y)$, then $\rho(img_q,\ img_x) \leq \rho(img_q,\ img_y)$. We also can take every image $img_i \in \mathcal{C}$ as a query image $img_q$, in order to obtain a set $\mathcal{R} = \{\sigma_1, \sigma_2,\ \ldots,\ \sigma_N\}$ of ranked lists for each image of the collection $\mathcal{C}$.

An image re-ranking algorithm is given by a function $f_r$, which takes a set of ranked lists $\mathcal{R}$ as the input and computes a new and more effective set of ranked lists $\hat{\mathcal{R}}$:

$$\hat{\mathcal{R}} = f_r(\mathcal{R}). \tag{1}$$

The image re-ranking algorithm, presented in next section, represents an implementation of the function $f_r$.

## 4. Image Re-Ranking Algorithm

The proposed re-ranking algorithm exploits the rich contextual information encoded in ranked lists, aiming at improving the effectiveness of CBIR systems. The algorithm is based on a recently proposed unsupervised strategy [29] that iteratively computes the similarity of top-$k$ lists. The main novelty of our approach consists in the use of a subset of the ranked lists (instead of using the complete distance matrix) and indexing structures for computing them, which enables the use of our re-ranking algorithm in large collections. In this way, besides the effectiveness gains, the efficiency and scalability issues are also addressed.

The central reasoning behind our image re-ranking 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* [29]. In general, if two images are similar, their ranked lists should be similar as well [26]. It is somehow close to the cluster hypothesis [32], which states that *"closely associated documents tend to be relevant to the same requests"*.

The modeling of contextual information considering only the similarity between ranked lists represents an advantage of our strategy. Instead of using the distance information, the proposed method requires only the ranking information. Since there are several image descriptors available and each one uses different approaches for distance computation, scores computed by different image descriptors usually are in different scales and requires normalization procedures. These variations can affect the effectiveness of the re-ranking approaches. On the other hand, even different approaches for distance computation produce ranked lists with the same structure. In this scenario, the proposed re-ranking method can be used for different CBIR tasks and can be easily adapted for other information retrieval tasks (*e.g.*, text or multimodal retrieval). Beyond that, the re-ranking method can be extended for using different similarity/dissimilarity measures among ranked lists, a well-established research area [12, 47, 48].

### *4.1. Contextual Ranked Lists*

In this section, we define the image re-ranking algorithm in terms of *contextual top-k lists*. The images at the top positions of ranked lists often are the most relevant images, in the sense that they usually represent the results in which users are interested. Therefore, the top-$k$ lists represent, by itself, a contextual description of images with respect to the whole dataset. In this scenario, we conjecture that, given any two images, and their respective top-$k$ lists, a new and more effective ranked list can be computed, which we named as *contextual ranked list*. Once new ranked lists are computed, the process can be iteratively repeated, representing the basis of our re-ranking algorithm.

The reasoning behind *contextual ranked lists* relies on exploiting the co-occurrence of similar images in the top-$k$ lists. Usually, the top positions of ranked lists contain many images that are similar to the query image and some *wrong* (non-similar) images. Those images placed at top positions usually are similar to each other and, therefore, there are many images in common in their ranked lists. The objective of the proposed re-ranking algorithm is to move the non-similar images down in the ranked lists, and, as a result of this process, improve the quality of ranked lists.

In the following, we formally define the contextual ranked lists. First, we use the definition of top positions of a ranked list as a *top-k list*, according to [12]. Let us consider the neighborhood set $\mathcal{N}(i,k)$ of an image $img_i$, which contains the $k$ most similar images to $img_i$ according to the ranked list $\sigma_q(i)$ computed by an indexing structure. A *top-k list* $\tau_i$ is a bijection from a domain $\mathcal{N}(i,k)$ (the members of the top $k$ list) to $[k] = \{1, 2, \ldots, k\}$. We say that $img_j$ appears in the top-$k$ list $\tau_i$ if $img_j \in \mathcal{N}(i,k)$. We interpret $\tau_i(j)$ as the position (or rank) of image $img_j$ in $\tau_i$.

For computing the contextual ranked lists, we define a rank-based distance measure $r_d$ based on the similarity of top-$k$ list. Assume that $\tau_i$ and $\tau_j$ are top-$k$ lists computed for images $img_i$ and $img_j$, respectively. Several similarity (or dissimilarity) measures for comparing $\tau_i$ and $\tau_j$ can be defined [12, 47, 48]. Let $d(\tau_i, \tau_j, k)$ denote a given distance measure for comparing top-$k$ lists, we define a non-iterative contextual distance measure $r_d(img_i, img_j)$ based on comparison of the top-$k$ lists, as follows:

$$r_d(img_i, img_j) = d(\tau_i, \tau_j, k). \tag{2}$$

Based on the conjecture that the rank-based distance measure $r_d$ represents a more effective distance between images, we can perform a re-ranking computing new ranked lists based on this measure. Let $\sigma_q^{(0)}(x)$ denotes the contextual ranked lists produced after this first re-ranking, we can say that, if $r_d(img_q, img_x) \leq r_d(img_q, img_y)$, then $\sigma_q^{(0)}(x) < \sigma_q^{(0)}(y)$, that is, $img_x$ is ranked before $img_y$ in the ranked list of $img_q$.

Since both input and output of the re-ranking process are ranked lists, this process can be repeated in an iterative manner. Let $^{(t)}$ be a superscript that denotes the iteration. Let $\tau_i^{(t)}$ be the top-$k$ list for image $img_i$ at iteration $t$, which is computed considering the rank-based distance measure $r_d^{(t)}$. Let $r_d^{(0)}$ be the rank-based distance at first iteration, we can define an iterative distance as follows:

$$r_d^{(t+1)}(img_i, img_j) = d(\tau_i^{(t)}, \tau_j^{(t)}, k). \tag{3}$$

Once the effectiveness of the rank-based distance measure improves along iterations, the effectiveness of ranked lists also improves. 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 this way, a larger $k$ can be considered for computation of top-$k$ lists along iterations, as follows:

$$r_d^{(t+1)}(img_i, img_j) = d(\tau_i^{(t)}, \tau_j^{(t)}, k+t). \tag{4}$$

After a given number of $T$ iterations, a final ranked-based distance $r_d^{(T)}$ is computed. Therefore, a definitive contextual ranked list $\sigma_q^T(x)$ can also be computed based on this distance. Let $\sigma_q^{(T)}(x)$ denotes the definitive contextual ranked lists produced, we can say that, if $r_d^{(T)}(img_q, img_x) \leq r_d^{(T)}$ $(img_q,\ img_y)$, then $\sigma_q^{(T)}(x) < \sigma_q^{(T)}(y)$.

Finally, we can obtain a contextual ranked list $\sigma_q^{(T)}$ for each $img_q \in \mathcal{C}$, computing a new set of ranked lists $\mathcal{R}^{(T)}$ and completing the re-ranking process.

### *4.2. Distance Measure Between Top-k Lists*

An approach to define the distance between two top-$k$ lists $\tau_i$ and $\tau_j$ proposed in [12] is to capture the extent of overlap between $\tau_i$ and $\tau_j$. This idea of overlap can be extended by considering not only the overlap at depth $k$, but also the cumulative overlap at increasing depths [29, 47]. For each $k_c \in \{1 \ldots k\}$, it is computed the overlap at $k_c$, and then those overlaps are averaged to derive a similarity measure. The measure gives higher weights to the first positions of top $k$ lists, which are considered many times. Equation 5 formally defines the intersection similarity measure $\delta$:

$$\delta(\tau_i, \tau_j, k) = \frac{\sum_{k_c=1}^{k} \mid \mathcal{N}(i, k_c) \cap \mathcal{N}(j, k_c) \mid}{k}. \tag{5}$$

Note that if two ranked lists present the same images at the first positions, the size of the intersection set is greater, and the value of $\delta$ is greater as well.

Since we are interested in a distance measure between top-$k$ lists, we define $d_\delta$ as follows:

$$d_\delta(\tau_i, \tau_j, k) = \frac{1}{1 + \delta(\tau_i, \tau_j, k)}. \tag{6}$$

### *4.3. The Image Re-Ranking Algorithm*

This section describes the image re-ranking algorithm, based on the presented ranked-based distance measure and the contextual ranked lists. The main input of the algorithm consists of a set of ranked lists $\mathcal{R}$, computed by indexing structures (discussed in details in Section 5). The size of ranked lists retrieved by the indexing structures is given by the parameter $N_S$. A trade-off control between effectiveness and efficiency can be obtained by varying this parameter. Increasing the $N_S$ value, higher effectiveness gains can be obtained by the re-ranking algorithm. On the other hand, more computational efforts are also required.

Given an initial set of ranked lists, an iterative approach is proposed. Let the superscript $^{(t)}$ denotes the current iteration, a new and more effective set of ranked lists $\mathcal{R}^{(t+1)}$ is computed by taking into account distances among top-$k$ lists. Next, $\mathcal{R}^{(t+1)}$ is used for the next execution of our re-ranking algorithm and so on. These steps are repeated along several iterations aiming to improve the effectiveness incrementally. After a number $T$ of iterations a definitive re-ranking is performed. Algorithm 1 outlines the proposed image re-ranking algorithm.

---

**Algorithm 1** Index-Based Image Re-Ranking Algorithm

---

**Require:** Image collection $\mathcal{C}$, parameters $k_s$, $T$, and $N_S$<br>
**Ensure:** Processed set of ranked lists $\mathcal{R}^{(T)}$<br>
$\mathcal{R}^{(0)} \leftarrow \varnothing$<br>
**for all** $img_i \in \mathcal{C}$ **do**<br>
    $\sigma_i \leftarrow computeTopKListByIndexing(img_i, N_S)$<br>
    $\mathcal{R}^{(0)} \leftarrow \mathcal{R}^{(0)} \cup \sigma_i$<br>
**end for**<br>
$t \leftarrow 0$<br>
$k \leftarrow k_s$<br>
**while** $t < T$ **do**<br>
    **for all** $\sigma_i \in \mathcal{R}^{(t)}$ **do**<br>
        **for all** $img_j \in \sigma_i$ **do**<br>
            **if** $\delta(\tau_i, \tau_j, k) \geq 0$ **then**<br>
                $r_d^{(t+1)}(i,j) \leftarrow d_\delta(\tau_i, \tau_j, k)$<br>
            **else**<br>
                $r_d^{(t+1)}(i,j) \leftarrow 1 + \sigma_i(j)$<br>
            **end if**<br>
        **end for**<br>
    **end for**<br>
    $\mathcal{R}^{(t+1)} \leftarrow reSortRankedLists(r_d^{(t+1)})$<br>
    $k \leftarrow k + 1$<br>
    $t \leftarrow t + 1$<br>
**end while**

---

The first loop (lines 1-5) calls the indexing structures aiming at retrieving the set of ranked lists $\mathcal{R}$, as defined in Section 3. The function *computeTopKListByIndexing* can use any indexing structure that can be parameterized to produce ranked lists, which are is computed using $kNN$ queries, considering the entire collection as query images. As mentioned before, the size of the ranked lists is given by the parameter $N_S$.

Note that the size of the top-$k$ lists starts with the value of the parameter $k_s$. At each iteration $t$, we increment the number of $k$ neighbors to be considered (line 19). The motivation behind this increment relies on the fact that the effectiveness of the ranked lists increase along iterations. In this way, non-relevant images are moved out from the first positions of the ranked lists and $k$ can be increased for considering more images.

It is also important to emphasize the motivation of the conditional statement in Line 11. The similarity between two top-$k$ lists, given by the function $\delta$, can return a score equals to zero. In these situations, the tie break criterion is based on position of the image in the ranked list of the previous iteration.

Figure 1 illustrates the overall searching process of the proposed approach, using re-ranking and indexing structures. The main characteristic of the system is the set of ranked lists, which represents the interface between the indexing structures and the re-ranking algorithm.

## 5. Indexing Structures

The problem of supporting nearest neighbor and range queries in metric spaces has recently attracted the attention of researchers. An excellent survey of metric access methods can be found in [8].

The pioneering work of Burkhard and Keller [7] provided two interesting techniques for partitioning a metric dataset in a recursive fashion. Their first approach partitions a dataset by choosing a representative from the set and grouping the objects with respect to their distance to the representative. The second approach divides the original set into a fixed number of groups and chooses a representative from each of the groups.

The metric tree of Uhlmann [44] and the Vantage-point tree (VP-tree) [54] are somewhat similar to the first technique of [7] as they divide the dataset into disjoint partitions according to a representative, called a "vantage point". In order to reduce the number of distance calculations to answer similarity queries using the VP-tree, Baeza-Yates et al. [5] suggested to use the same vantage point in all partitions that belong to the same level. Then, a binary tree degenerates into a simple list of vantage points. Bozkaya and Özsoyogly [6]

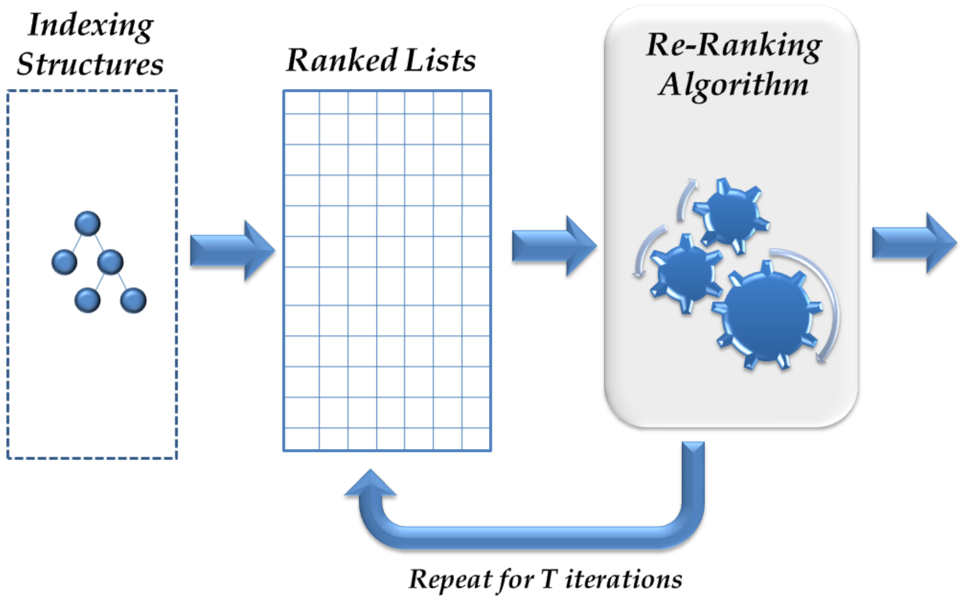


Figure 1: Overall searching process of a CBIR system using re-ranking and indexing structures.

proposed an extension of the VP-tree called the Multi-Vantage-Point tree (MVP-tree), which carefully chooses $m$ vantage points for each level of the tree. The Generalized Hyperplane tree (GH-tree) [44] is another method that recursively divides the dataset into disjoint partitions by selecting objects as representatives and assigning the remaining ones to the closest representative.

The Metric tree (M-tree) [9] is a height-balanced tree also based on the second technique presented in [7], which stores the data in the leaves and builds an appropriate cluster hierarchy on top, allowing for dynamic operations. Traina Jr. et al. [43] proposed an extension of the M-tree, named Slim-tree. They introduced three new features: (1) a node-splitting strategy based on the MST (minimum spanning tree) algorithm, (2) an insertion policy based on the node occupancy, and (3) a post-processing algorithm to reduce the overlapping volumes in the tree, called Slim-down. Vieira et al. [45] suggested to relax the height balance constraint by keeping a trade-off between breadth-searching and depth-searching in order to reduce the overlapping between nodes in high-density regions, improving the search performance in those regions.

The Ball-and-Plane tree (BP-tree) [3] combines the advantages of both the first (partitions) and the second (groups) technique of [7] in order to achieve a structure of tight and low overlapping clusters, yielding significantly improved performance on performing similarity search.

## 6. Experimental Evaluation

In this section, we analyze the performance and the scalability of our technique. Our experiments are intended (i) to validate that, even considering scalable data structures (only a subset of ranked lists computed by indexing structures) as the input, our re-ranking algorithm can obtain significant gains in effectiveness and efficiency; (ii) to demonstrate that our approach does not depend on a specific image descriptor or indexing structure; and (iii) to show that the proposed method scales up very well and, hence, it is suitable for large collections.

### *6.1. Experimental Setup*

Experiments were conducted on a large set of images, known as the Amsterdam Library of Object Images (ALOI)[1] [15]. It is a collection of 72,000 images from 1,000 classes of objects, with a common background and different viewpoint, occlusion, and illumination.

We tested our approach with five image descriptors described in literature and extensively used by the computer vision and image processing communities. Those approaches have been used for the convenience of obtaining large datasets in which a reasonable ground truth can be established. Regardless of that, our re-ranking scheme itself does not use any property related to the nature of the methods. Our image descriptors are the following:

- Auto Color Correlation (ACC) [17];
- Border/Interior pixel Classification (BIC) [39];
- Color Coherence Vectors (CCV) [25];
- Global Color Histogram (GCH) [40];
- Local Color Histograms (LCH) [23].

[1] `http://staff.science.uva.nl/~aloi/` As of May 2012.

For our experimental evaluation, we adopted the implementation of M-tree, Slim-tree, and DBM-tree from the GBDI Arboretum Library[2]. It offers a robust and uniform platform in which one can perform a reliable comparative analysis between different metric access methods. For that reason, BP-tree was implemented into the GBDI Arboretum Library, with the same code optimization. In order to guarantee a fair comparison, all of the compared methods were configured using their best recommended setup. A performance comparison between those indexing structures can be found in [3]. Our interest here is to use a well-established validation framework in which we can perform a behavior analysis of the proposed method with respect to indexing structures. Regardless of that, our approach is flexible and, hence, any other indexing method can be used.

As described in Section 4, the image re-ranking algorithm relies on three parameters: $k_s$ (the initial size of top-$k$ lists); $N_S$ (the size of ranked lists retrieved by the indexing structures); and $T$ (the number of iterations). The parameters used in our experiments were: $k_s$ equals to 45; $N_S$ set to 7,200; and $T$ equals to 1. The parameter $N_S$ was defined as 10% of the dataset size. The parameters $k_s$ and $T$ were established through experimental tests according to an approach used in [29]. Retrieval scores are computed ranging the parameters $k_s$ in the interval [1, 60] (in steps of 5) and $T$ in the interval [1, 5]; and the best parameter values are determined. In these experiments, we considered only the combination BPTree + ACC. The same parameter values are used for all the possible combinations between the image descriptors and the indexing structures.

We assess the effectiveness of each of those combinations using the metrics of *Precision* and *Recall*. Precision is the ratio of the number of relevant images retrieved to the total number of images retrieved. Recall is the ratio of the number of relevant images retrieved to the total number of relevant images in the database. However, there is a trade-off between Precision and Recall. Greater Precision decreases Recall and greater Recall leads to decreased Precision. For that reason, we choose the *Average Precision* (AP) as the metric used for assessing the effectiveness of CBIR tasks. The Average Precision combines Recall and Precision into a single measure by taking the set of ranks at which the relevant images occur, computing the precision at those positions, and then averaging the set of precision values obtained. The average precision across a series of queries can be averaged, resulting in a measure known as *Mean Average Precision* (MAP).

[2] `http://www.gbdi.icmc.usp.br/arboretum/` As of May 2012.

### 6.2. *Effectiveness Analysis*

Tables 1, 2, 3, and 4 present the MAP obtained for the image descriptors before and after the execution of the re-ranking algorithm on ranked lists produced by BP-tree, DBM-tree, M-tree, and Slim-tree, respectively.

Table 1: MAP obtained for different image descriptors using ranked lists produced by BP-tree.

| **Image Descriptor** | **Before Re-Ranking** | **After Re-Ranking** | **Effectiveness Gain** |
|---|---|---|---|
| ACC | 44.15% | 46.12% | 4.44% |
| BIC | 71.95% | 78.84% | 9.57% |
| CCV | 47.77% | 50.96% | 6.68% |
| GCH | 50.87% | 53.14% | 4.47% |
| LCH | 58.85% | 66.03% | 12.19% |

Table 2: MAP obtained for different image descriptors using ranked lists produced by DBM-tree.

| **Image Descriptor** | **Before Re-Ranking** | **After Re-Ranking** | **Effectiveness Gain** |
|---|---|---|---|
| ACC | 44.17% | 46.14% | 4.47% |
| BIC | 71.90% | 78.87% | 9.70% |
| CCV | 47.77% | 50.99% | 6.73% |
| GCH | 50.88% | 53.20% | 4.55% |
| LCH | 58.84% | 65.94% | 12.08% |

Table 3: MAP obtained for different image descriptors using ranked lists produced by M-tree.

| **Image Descriptor** | **Before Re-Ranking** | **After Re-Ranking** | **Effectiveness Gain** |
|---|---|---|---|
| ACC | 44.17% | 46.15% | 4.47% |
| BIC | 71.88% | 78.96% | 9.85% |
| CCV | 47.79% | 51.01% | 6.74% |
| GCH | 50.89% | 53.19% | 4.53% |
| LCH | 58.84% | 65.91% | 12.03% |

The results indicate that the proposed method improves the effectiveness of CBIR tasks. As we can observe, our approach provides significant

Table 4: MAP obtained for different image descriptors using ranked lists produced by Slim-tree.

| **Image Descriptor** | **Before Re-Ranking** | **After Re-Ranking** | **Effectiveness Gain** |
|---|---|---|---|
| ACC | 44.18% | 46.16% | 4.48% |
| BIC | 71.91% | 79.00% | 9.86% |
| CCV | 47.78% | 51.00% | 6.75% |
| GCH | 50.90% | 53.21% | 4.54% |
| LCH | 58.82% | 65.97% | 12.16% |

effectiveness gains, ranging from 4.44% to 12.19%. For instance, the use of BP-Tree + LCH produced the higher effectiveness gain of all the possible combinations between the image descriptors and the indexing structures, yielding an improvement equals to **12.19%**.

Paired $t$-tests were performed to verify the statistical significance of those results. For that, the confidence intervals for the differences between paired means of each class from the database were computed to compare every pair of approaches. If the confidence interval includes zero, the difference is not significant at that confidence level. If the confidence interval does not include zero, then the sign of the difference indicates which alternative is better.

Tables 5, 6, 7, and 8 present the confidence intervals (with a confidence of 99.9%) of the differences between the MAP obtained for the image descriptors before and after the execution of the re-ranking algorithm on ranked lists produced by BP-tree, DBM-tree, M-tree, and Slim-tree, respectively.

Table 5: Differences between MAP obtained for the image descriptors before and after using the image re-ranking algorithm on ranked lists produced by BP-tree, at a confidence of 99.9%.

| **Image** | | **Confidence Interval (99.9%)** | |
|---|---|---|---|
| **Descriptor** | **Mean** | **min.** | **max.** |
| ACC | 1.96% | 1.35% | 2.57% |
| BIC | 6.89% | 6.11% | 7.67% |
| CCV | 3.19% | 2.59% | 3.79% |
| GCH | 2.27% | 1.70% | 2.84% |
| LCH | 7.18% | 6.23% | 8.12% |

Since the confidence intervals do not include zero in any case, those results confirm that the proposed method improves the effectiveness of CBIR

Table 6: Differences between MAP obtained for the image descriptors before and after using the image re-ranking algorithm on ranked lists produced by DBM-tree, at a confidence of 99.9%.

| **Image Descriptor** | **Mean** | **Confidence Interval (99.9%)** min. | max. |
|---|---|---|---|
| ACC | 1.98% | 1.37% | 2.58% |
| BIC | 6.97% | 6.19% | 7.75% |
| CCV | 3.22% | 2.61% | 3.82% |
| GCH | 2.31% | 1.75% | 2.88% |
| LCH | 7.11% | 6.17% | 8.04% |

Table 7: Differences between MAP obtained for the image descriptors before and after using the image re-ranking algorithm on ranked lists produced by M-tree, at a confidence of 99.9%.

| **Image Descriptor** | **Mean** | **Confidence Interval (99.9%)** min. | max. |
|---|---|---|---|
| ACC | 1.98% | 1.37% | 2.58% |
| BIC | 7.08% | 6.30% | 7.86% |
| CCV | 3.22% | 2.62% | 3.82% |
| GCH | 2.30% | 1.73% | 2.87% |
| LCH | 7.08% | 6.14% | 8.01% |

Table 8: Differences between MAP obtained for the image descriptors before and after using the image re-ranking algorithm on ranked lists produced by Slim-tree, at a confidence of 99.9%.

| **Image Descriptor** | **Mean** | **Confidence Interval (99.9%)** min. | max. |
|---|---|---|---|
| ACC | 1.98% | 1.37% | 2.29% |
| BIC | 7.09% | 6.31% | 7.87% |
| CCV | 3.23% | 2.63% | 3.83% |
| GCH | 2.31% | 1.74% | 2.88% |
| LCH | 7.15% | 6.21% | 8.10% |

tasks, independent of the image descriptor and/or the indexing structure employed to produce ranked lists.

### *6.3. Efficiency Analysis*

In a sequential approach, retrieving a ranked list requires $N$ distance calculations (with $N$ equals to 72,000 for the ALOI dataset). The use of indexing structures reduce significantly the need for distance calculations, which impacts on both efficiency and scalability of the CBIR systems.

Table 9 presents, for each combination between the image descriptors and the indexing structures, the average number of distance calculations and the efficiency gains obtained by using an indexing structure regarding the linear scan. As we can observe, the use of indexing structures reduces up to 73% (for ACC + BP-Tree) the average number of distance calculations required for producing ranked lists.

Table 9: Average number of distance calculations performed for different indexing structures.

| **Indexing Structure** | **Image Descriptor** | **Distance Calculations** | **Efficiency Gain** |
|---|---|---|---|
| BP-tree | ACC | 19358.40 | 73.11% |
| | BIC | 27821.20 | 61.36% |
| | CCV | 30846.80 | 57.16% |
| | GCH | 23264.80 | 67.69% |
| | LCH | 42896.50 | 40.42% |
| DBM-tree | ACC | 22340.50 | 68.97% |
| | BIC | 33152.70 | 53.95% |
| | CCV | 38243.40 | 46.88% |
| | GCH | 31315.40 | 56.51% |
| | LCH | 47518.00 | 34.00% |
| M-tree | ACC | 28244.00 | 60.77% |
| | BIC | 43846.90 | 39.10% |
| | CCV | 47014.70 | 34.70% |
| | GCH | 39654.90 | 44.92% |
| | LCH | 55672.50 | 22.68% |
| Slim-tree | ACC | 32702.20 | 54.58% |
| | BIC | 40835.80 | 43.28% |
| | CCV | 53857.30 | 25.20% |
| | GCH | 52624.10 | 26.91% |
| | LCH | 57574.00 | 20.04% |

Considering the re-ranking step, the main contribution of the proposed method refers to the use of only a subset of the ranked lists ($N_S$ equals to 7,200). This characteristic impacts drastically on the scalability of CBIR systems, since using the complete ranked lists or the complete distance matrix can be impracticable ($N^2$=5,184,000,000).

The average time required for computing the re-ranking for each ranked list is $0.06s$, considering the parameters settings described in Section 6.1. We used a Linux Ubuntu 10.04, running on a Intel Xeon X7560 CPU and a C implementation. It is important to realize that the overall efficiency of the proposed retrieval system can be even improved, if we consider the use of possible optimizations based on exploiting parallel architectures in the implementation of the re-ranking method.

### *6.4. Scalability Analysis*

In this section, we evaluate the scalability of the image re-ranking algorithm. While indexing structures save efforts in computing distances, the re-ranking algorithm is also prepared for dealing with growing ranked lists.

The size of the ranked lists analyzed is given by the parameter $N_S$ and, therefore, the asymptotic complexity of the algorithm is $O(N)$. This parameter represents an important trade-off control between effectiveness and efficiency.

In the following, we present a set of experiments aiming at evaluating the impact of the parameter $N_S$ on the results. We ranged the parameter from 70 to 7000, reporting for each descriptor: $(i)$ the average time (in seconds) of re-ranking by ranked list; and $(ii)$ the effectiveness gain obtained. In these experiments, we considered only BP-tree for producing the ranked lists, as it achieved the best performance among all the indexing structures.

Figure 2 shows the impact of the parameter $N_S$ (size of ranked lists) on the effectiveness gains. A quickly grow of the effectiveness gains was obtained for small values of $N_S$ and a stabilization can be observed for larger values (specially for $N_S \geq 2800$). All descriptors presented analogous results.

The impact of parameter $N_S$ on the average time of re-raking by ranked list is illustrated in Figure 3. A linear growth of average times can be observed for all image descriptors.

As we can observe in Figures 2 and 3, by increasing the size of the ranked lists, the proposed method exhibits a linear increasing of the average time, while the effectiveness increase very quickly at small values of $N_S$. This behavior represents the most important advantage of our method in comparison with other re-ranking methods, allowing its use for large datasets.

While other state-of-the-art approaches [51, 46] present computing cost of $O(N^3)$ and storage requirements of $O(N^2)$, our method is linear both for computational and storage costs.

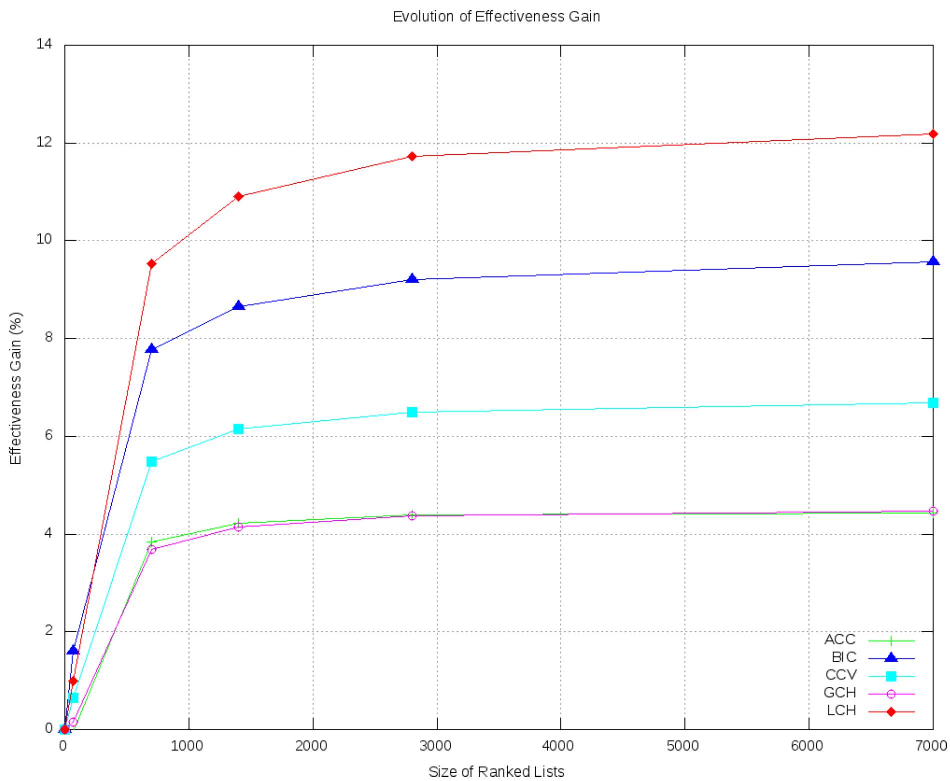


Figure 2: Impact of the size of the ranked lists on effectiveness gain.

### *6.5. Comparison to Other Approaches*

We also evaluate our method in comparison with other state-of-the-art post-processing methods. We use the MPEG-7 [20] dataset, with the called bullseye score (*Recall@40*), commonly used for post-processing methods evaluation and comparison. The following shape descriptors, also used by other methods, were considered: Inner Distance Shape Context (IDSC) [21], Contour Features Descriptor (CFD) [26], Aspect Shape Context (ASC) [22], and Articulation-Invariant Representation (AIR) [16]. Table 10 presents results of our *Index-Based Image Re-Ranking* algorithm. Despite of low computational efforts required and the scalable behavior previous discussed, the proposed re-ranking method presented effectiveness results comparable to state-of-the-art approaches.

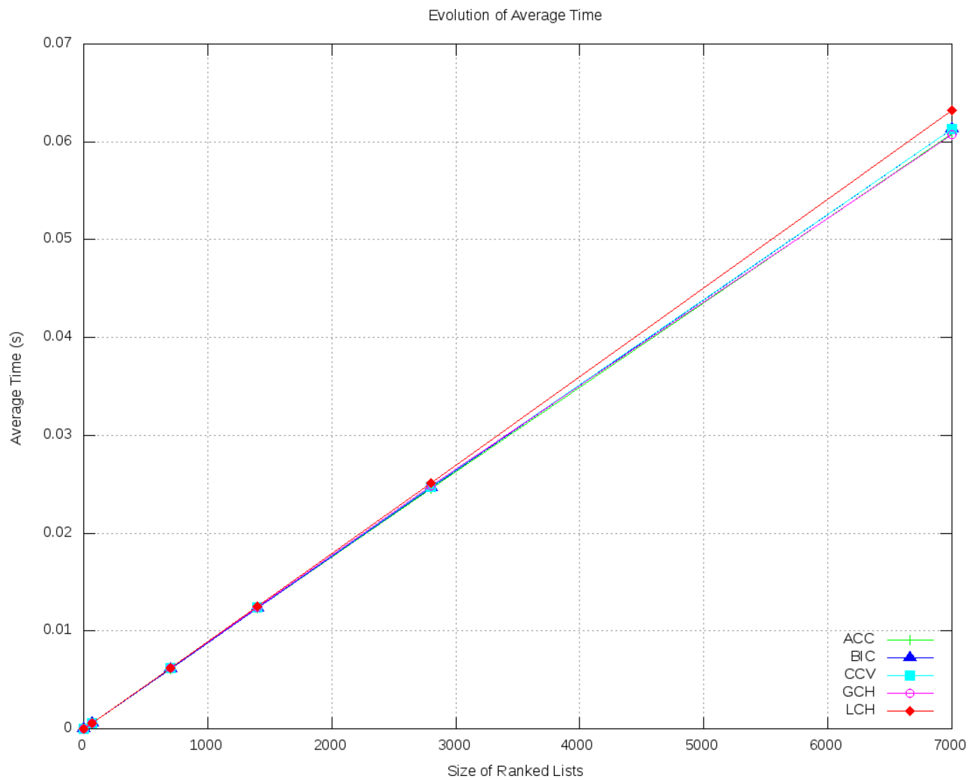


Figure 3: Impact of the size of the ranked lists on average time.

### *6.6. Impact of Indexing Structures*

We also evaluate the impact of the use of indexing structures on the effectiveness results. Table 11 presents the retrieval scores (*Recall@40*) for the four shape descriptors considered, considering two different values of $N_S$. We also compared with the results obtained by the re-ranking algorithm without the indexing structures (considering the full distance matrix). We can observe that, in general, the loss in terms of effectiveness is very low in comparison with the relevant gains in scalability. For the AIR descriptor, which presents a high precision (similar images at beginning of ranked lists), the indexing approach overcomes the original algorithm.

## 7. Conclusions

In this paper, we have presented a scalable re-ranking method that exploits contextual information for improving the effectiveness of CBIR tasks.

Table 10: Post-processing methods comparison on MPEG-7 dataset *(Recall@40)*.

| **Algorithm** | **Shape Descriptor** | **Score** | **Gain** |
|---|---|---|---|
| **Shape Descriptors** | | | |
| Contour Features Descritpor (CFD) [26] | - | 84.43% | - |
| Inner Distance Shape Context (IDSC) [21] | - | 85.40% | - |
| Aspect Shape Context (ASC) [22] | - | 88.39% | - |
| Articulation-Invariant Rep. (AIR) [16] | - | 93.67% | - |
| **Post-Processing Methods** | | | |
| Graph Transduction (LP) [49] | IDSC | 91.00% | +6.56% |
| **Index-Based Image Re-Ranking** | **IDSC** | **91.56%** | **+7.21%** |
| **Index-Based Image Re-Ranking** | **CFD** | **92.85%** | **+9.97%** |
| Contextual Spaces [28] | CFD | 93.02% | +10.17% |
| Locally Constrained Diffusion Process [50] | IDSC | 93.32% | +9.27% |
| Shortest Path Propagation [46] | IDSC | 93.35% | +9.31% |
| **Index-Based Image Re-Ranking** | **ASC** | **94.09%** | **+6.45%** |
| Locally Constrained Diffusion Process [50] | ASC | 95.96% | +8.56% |
| **Index-Based Image Re-Ranking** | **AIR** | **99.93%** | **+6.68%** |
| Tensor Product Graph [52] | AIR | 99.99% | +6.75% |

Table 11: Retrieval scores (*Recall@40*) for different descriptors, indexing structures and sizes of ranked lists ($N_S$).

| | DBM-tree | | M-tree | | Slim-tree | | BP-tree | | Without |
|---|---|---|---|---|---|---|---|---|---|
| | $N_S$=140 | $N_S$=280 | $N_S$=140 | $N_S$=280 | $N_S$=140 | $N_S$=280 | $N_S$=140 | $N_S$=280 | Indexing |
| IDSC | 90.56% | 91.56% | 90.56% | 91.56% | 90.56% | 91.56% | 90.56% | 91.56% | 92.18% |
| CFD | 91.15% | 92.85% | 91.15% | 92.85% | 91.15% | 92.85% | 91.15% | 92.85% | 94.13% |
| ASC | 93.06% | 94.09% | 93.06% | 94.09% | 93.06% | 94.09% | 93.06% | 94.09% | 94.69% |
| AIR | 99.93% | 99.93% | 99.93% | 99.93% | 99.93% | 99.93% | 99.93% | 99.93% | 99.90% |

The main idea consists in analyzing the similarity between ranked lists for performing a re-ranking process.

Different from previous works, the proposed re-ranking method does not require distance information among all the images of a given collection or complete ranked lists. Instead, our technique relies on ranked lists produced by efficient indexing structures. Such a strategy makes it scalable and, hence, well-suited to large datasets.

We have conducted a large set of experiments on a well-known and public dataset, considering several indexing structures. Experimental results have demonstrated that the proposed re-ranking method can achieve significant effectiveness gains (up to 12.19% better) and, at the same time, improve considerably the efficiency (up to 73.11% faster).

Future work includes the evaluation of other CBIR descriptors (e.g., local features [2] or motion patterns [1]) and indexing structures. In addition, the proposed re-ranking algorithm can be extended for combining results obtained from different CBIR descriptors (rank aggregation tasks). Finally, we want to investigate the effects of using parallel computing for accelerating the image re-ranking algorithm.

## 8. Acknowledgments

This research was partially supported by AMD, Microsoft Research and Brazilian agencies FAPESP (Grants 2007/52015-0, 2009/05951-8, 2009/18438-7, 2011/11171-5, and 2013/08645-0), CNPq (Grants 306580/2012-8 and 484254/2012-0), and CAPES.

## References

[1] Almeida, J., Leite, N.J., Torres, R.S.. Comparison of video sequences with histograms of motion patterns. In: IEEE Int. Conf. Image Processing (ICIP'11). 2011. p. 3673–3676.

[2] Almeida, J., Rocha, A., Torres, R.S., Goldenstein, S.. Making colors worth more than a thousand words. In: ACM Int. Symp. Applied Computing (ACM-SAC'08). 2008. p. 1180–1186.

[3] Almeida, J., Torres, R.S., Leite, N.J.. BP-tree: An efficient index for similarity search in high-dimensional metric spaces. In: ACM Int. Conf. Information and Knowledge Management (CIKM'10). 2010. p. 1365–1368.

[4] Almeida, J., Valle, E., Torres, R.S., Leite, N.J.. DAHC-tree: An effective index for approximate search in high-dimensional metric spaces. Journal of Information and Data Management 2010;1(3):375–390.

[5] Baeza-Yates, R.A., Cunto, W., Manber, U., Wu, S.. Proximity matching using fixed-queries trees. In: Annual Symp. Combinatorial Pattern Matching (CPM'94). volume 807 of *Lecture Notes in Computer Science*; 1994. p. 198–212.

[6] Bozkaya, T., Özsoyoglu, Z.M.. Indexing large metric spaces for similarity search queries. ACM Transactions on Database Systems 1999;24(3):361–404.

[7] Burkhard, W.A., Keller, R.M.. Some approaches to best-match file searching. Communications of the ACM 1973;16(4):230–236.

[8] Chávez, E., Navarro, G., Baeza-Yates, R.A., Marroquín, J.L.. Searching in metric spaces. ACM Computing Surveys 2001;33(3):273–321.

[9] Ciaccia, P., Patella, M., Zezula, P.. M-tree: An efficient access method for similarity search in metric spaces. In: Int. Conf. Very Large Data Bases (VLDB'97). 1997. p. 426–435.

[10] Datta, R., Joshi, D., Li, J., Wang, J.Z.. Image retrieval: Ideas, influences, and trends of the new age. ACM Computing Surveys 2008;40(2):5:1–5:60.

[11] Elmasri, R.A., Navathe, S.B.. Fundamentals of Database Systems. Addison-Wesley Longman Publishing Co., Inc., 2005.

[12] Fagin, R., Kumar, R., Sivakumar, D.. Comparing top k lists. In: ACM-SIAM Symposium on Discrete algorithms (SODA'03). 2003. p. 28–36.

[13] Ferreira, C.D., dos Santos, J.A., da S. Torres, R., Gonçalves, M.A., Rezende, R.C., Fan, W.. Relevance feedback based on genetic programming for image retrieval. Pattern Recogninion Letters 2011;32(1):27–37.

[14] Gaede, V., Günther, O.. Multidimensional access methods. ACM Computing Surveys 1998;30(2):170–231.

[15] Geusebroek, J.M., Burghouts, G.J., Smeulders, A.W.M.. The amsterdam library of object images. International Journal of Computer Vision 2005;61(1):103–112.

[16] Gopalan, R., Turaga, P., Chellappa, R.. Articulation-invariant representation of non-planar shapes. In: 11th European Conference on Computer Vision (ECCV'2010). volume 3; 2010. p. 286–299.

[17] Huang, J., Kumar, R., Mitra, M., Zhu, W.J., Zabih, R.. Image indexing using color correlograms. In: IEEE Int. Conf. Computer Vision and Pattern Recognition (CVPR'97). 1997. p. 762–768.

[18] Jagadish, H.V., Mendelzon, A.O., Milo, T.. Similarity-based queries. In: ACM SIGACT-SIGMOD-SIGART Symp. Principles of Database Systems (PODS'95). 1995. p. 36–45.

[19] Kontschieder, P., Donoser, M., Bischof, H.. Beyond pairwise shape similarity analysis. In: Asian Conference on Computer Vision. 2009. p. 655–666.

[20] 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 (CVPR'2000). 2000. p. 424–429.

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

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

[23] Lu, H., Ooi, B., Tan, K.. Efficient image retrieval by color contents. In: Int. Conf. Applications of Databases (ADB'94). volume 819 of *Lecture Notes in Computer Science*; 1994. p. 95–108.

[24] 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.

[25] Pass, G., Zabih, R., Miller, J.. Comparing images using color coherence vectors. In: ACM Int. Conf. Multimedia (ACM-MM'96). 1996. p. 65–73.

[26] 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 (VISAPP'2010). volume 1; 2010. p. 197 – 202.

[27] 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.

[28] Pedronette, D.C.G., da S. Torres, R.. Exploiting contextual spaces for image re-ranking and rank aggregation. In: ACM International Conference on Multimedia Retrieval (ICMR'11). 2011. p. 13:1–13:8.

[29] 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 (CAIP'2011). volume 6854; 2011. p. 369–376.

[30] 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 (CVPR'2009). 2009. p. 2358–2365.

[31] Ramakrishnan, R., Gehkre, J.. Database Management Systems. McGraw-Hill Co., Inc.,, 2003.

[32] van Rijsbergen, C.J.. Information Retrieval. London: Butterworth-Heinemann, 1979.

[33] Rocha, A., Almeida, J., Nascimento, M.A., Torres, R., Goldenstein, S.. Efficient and flexible cluster-and-search approach for cbir. In: Int. Conf. Advanced Concepts for Intelligent Vision Systems (ACIVS'08). volume 5259 of *Lecture Notes in Computer Science*; 2008. p. 77–88.

[34] Rui, Y., Huang, T., Ortega, M., Mehrotra, S.. Relevance feedback: a power tool for interactive content-based image retrieval. IEEE Transactions on Circuits and Systems for Video Technology 1998;8(5):644–655.

[35] Rui, Y., Huang, T.S., Chang, S.F.. Image retrieval: Current techniques, promising directions, and open issues. Journal of Visual Communication and Image Representation 1999;10(1):39–62.

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

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

[38] 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 (VISAPP'2010). volume 1; 2010. p. 118–122.

[39] Stehling, R.O., Nascimento, M.A., Falcão, A.X.. A compact and efficient image retrieval approach based on border/interior pixel classification. In: ACM Int. Conf. Information and Knowledge Management (CIKM'02). 2002. p. 102–109.

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

[41] Tao, D., Tang, X., Li, X., Wu, X.. Asymmetric bagging and random subspace for support vector machines-based relevance feedback in image retrieval. Pattern Analysis and Machine Intelligence, IEEE Transactions on 2006;28(7):1088–1099.

[42] Tian, X., Tao, D., Hua, X.S., Wu, X.. Active reranking for web image search. IEEE Transactions on Image Processing 2010;19(3):805–820.

[43] Traina Jr., C., Traina, A.J.M., Faloutsos, C., Seeger, B.. Fast indexing and visualization of metric data sets using slim-trees. IEEE Transactions on Knowledge and Data Engineering 2002;14(2):244–260.

[44] Uhlmann, J.K.. Satisfying general proximity/similarity queries with metric trees. Information Processing Letters 1991;40(4):175–179.

[45] Vieira, M.R., Traina Jr., C., Chino, F.J.T., Traina, A.J.M.. DBM-tree: Trading height-balancing for performance in metric access methods. Journal of the Brazilian Computer Society 2006;11(3):37–52.

[46] Wang, J., Li, Y., Bai, X., Zhang, Y., Wang, C., Tang, N.. Learning context-sensitive similarity by shortest path propagation. Pattern Recognition 2011;44(10-11):2367–2374.

[47] Webber, W., Moffat, A., Zobel, J.. A similarity measure for indefinite rankings. ACM Transactions on Information Systems 2010;28(4):20:1–20:38.

[48] Wu, S., Crestani, F.. Methods for ranking information retrieval systems without relevance judgments. In: ACM Symposium on Applied Computing (SAC'03). 2003. p. 811–816.

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

[50] 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 (CVPR'2009). 2009. p. 357–364.

[51] Yang, X., Latecki, L.J.. Affinity learning on a tensor product graph with applications to shape and image retrieval. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR'2011). 2011. p. 2369–2376.

[52] Yang, X., Prasad, L., Latecki, L.J.. Affinity learning with diffusion on tensor product graph. IEEE Transactions on Pattern Analysis and Machine Intelligence 2013;35(1):28–38.

[53] Yang, Y., Nie, F., Xu, D., Luo, J., Zhuang, Y., Pan, Y.. A multimedia retrieval framework based on semi-supervised ranking and relevance feedback. IEEE Transactions on Pattern Analysis and Machine Intelligence 2012;34(4):723–742.

[54] Yianilos, P.N.. Data structures and algorithms for nearest neighbor search in general metric spaces. In: ACM/SIGACT-SIAM Int. Symp. Discrete Algorithms (SODA'98). 1993. p. 311–321.

[55] Zezula, P., Amato, G., Dohnal, V., Batko, M.. Similarity Search: The Metric Space Approach. Springer-Verlag, Inc., 2005.