Exploiting Contextual Information for Image Re-Ranking and Rank Aggregation

Daniel Carlos Guimarães Pedronette and Ricardo da S. Torres

Abstract Content-based Image Retrieval (CBIR) systems aims to retrieve the most similar images in a collection, given a query image. Since users are interested in the returned images placed at the first positions of ranked lists (which usually are the most relevant ones), the effectiveness of these systems is very dependent on the accuracy of ranking approaches. This paper presents a novel re-ranking algorithm aiming to exploit *contextual information* for improving the effectiveness of rankings computed by CBIR systems. In our approach, ranked lists and distance scores are used to create *context images*, later used for retrieving contextual information. We also show that our re-ranking method can be applied to other tasks, such as: *(i)* combining ranked lists obtained by using different image descriptors (rank aggregation); and *(ii)* combining post-processing methods. Conducted experiments involving shape, color, and texture descriptors and comparisons with other post-processing methods demonstrate the effectiveness of our method.

Keywords content-based image retrieval · re-ranking, rank aggregation · image processing · contextual information

1 Introduction

The continuously decrease of storage devices costs and the technological improvements in image acquisition and sharing facilities have enabled the dissemination of very large digital image collections, accessible through various technologies. In this scenario, effective and efficient systems for searching and organizing these contents are of great interest.

Content-based image retrieval (CBIR) can be seen as any technology that helps to search and organize digital picture archives by means of their visual content [9]. In general, 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*. An image content descriptor is characterized by [33]: *(i)* an extraction algorithm that encodes image features into feature vectors; and *(ii)* a similarity measure used to compare two images. The similarity between two images is computed as a function of the distance of their feature vectors.

A direct way to improve the effectiveness of CBIR systems consists in using more accurate features for describing images. Another possibility is related to the definition of similarity (or distance) functions - that would be able to measure the distance between feature vectors in a more effective way.

Commonly, CBIR systems compute similarity considering only pairs of images. On the other hand, the user perception usually considers the query specification and the query responses in a given context. In interactive applications, the use of context can play an important role [1]. Context can be broadly defined as all information about the whole situation relevant to an application and its set of users. In information retrieval and recommendation systems, that includes geographic information, user profiles, and relationships among users and objects that can be used for improving the effectiveness of obtained results. In a CBIR sce-

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

nario, relationships among images, encoded in ranked lists, can be used for extracting contextual information.

In this paper, we present a new post-processing method that re-ranks images by taking into account *contextual information* encoded in ranked lists and distance among images. We propose a novel approach for retrieving contextual information, by creating a *gray scale image* representation of distance matrices computed by CBIR descriptors (referenced in this paper as *context image*). The context image is constructed for the k-nearest neighbors of a query image and analyzed using image processing techniques. The use of image processing techniques for *contextual information* representation and processing is an important novelty of our work. Our method uses distance matrices computed by CBIR descriptors that are later processed considering their image representation. The median filter, for instance, which is a well-known nonlinear filter often used for removing noise, is exploited in our approach to improve the quality of distance scores. Basically, we consider that "*wrong*" distances can be considered and represented as "*noise*" in the context image, and the median filter is used for filtering this noise out. In fact, a very large number of image processing techniques can be used for extracting useful information from context images. We believe that our strategy opens a new are of investigation related to the used of image processing approaches for analyzing distances computed by CBIR descriptor, in tasks such as image re-ranking, rank aggregation, and clustering.

We evaluated the proposed method on shape, color, and texture descriptors. Experimental results demonstrate that the proposed method can be used in several CBIR tasks, since it yields better results in terms of effectiveness performance than various post-processing algorithms recently proposed in the literature.

This paper differs from previous works [27, 30] with regard to the following aspects: *(i)* it presents and discusses in more details the main concepts of the proposed methods; *(ii)* it extends both re-ranking and rank aggregation algorithms by considering more contextual information and; *(iii)* it presents new experimental results that overcomes the original methods.

The paper is organized as follows: Section 2 discusses related work and Section 3 presents the problem definition. Section 4.1 describes the contextual information representation, while Sections 4.2 and 4.3 describe the re-ranking and rank aggregation methods, repectively. Section 4.4 discusses how to use our approach for combining post-processing methods. Experimental design and results are reported in Section 5. Finally, Section 6 presents conclusions and future work.

2 Related Work

This section discusses related work. Section 2.1 discusses the re-ranking approaches and Section 2.2 describes the rank aggregation methods.

2.1 Re-Ranking

Recently, several approaches have been proposed for performing re-ranking tasks on various information retrieval systems [32, 18, 10, 31, 36, 27, 41, 3]. In general, these methods perform a *post-processing analysis* that uses an initial ranking and exploits additional information (e.g., relationships among items, user profiles) for improving the effectiveness of ranked lists.

In the Information Retrieval scenario, the term "*global ranking*" was proposed in [32] for designating a ranking model that takes all the documents together as its input, instead of only individual objects. In other words, a global ranking uses not only the information of documents, but also the relation information among them.

The Continuous Conditional Random Fields (CRF) has been proposed in [32] for conducting the learning task in global ranking tasks. This model is defined as a conditional probability distribution over ranking scores of objects. It represents the content information of objects as well as the relation information between objects, necessary for global ranking. A global ranking framework that solves the problem via data fusion was proposed in [8]. The main idea of the approach is to take each retrieved document as a pseudo-information retrieval system. Each document generates a pseudo-ranked list by a global function. A data fusion algorithm is then adapted to generate the final ranked list.

Inter-documents similarity are considered in [10] and a clustering approach is applied for regularizing retrieval scores. In [45], a semi-supervised label propagation algorithm [50] was applied for re-ranking documents in information retrieval applications.

In the CBIR scenario, several methods also have been proposed for post-processing retrieval tasks, considering relationships among images. A graph transduction learning approach is introduced in [47]. The algorithm computes the shape similarity of a pair of shapes in the context of other shapes as opposed to considering only pairwise relations. The influence among shape similarities in an image collection is analyzed in [48]. 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 shortest path propagation

algorithm was proposed in [43], 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. Then a new distance measure is learned based on the shortest path that distance is used to replace the original distance measure. Another approach based on propagating the similarity information in a weighted graph is proposed in [46] and called of *affinity learning*. Instead of propagating the similarity information on the original graph, it uses a tensor product graph (TPG) obtained by the tensor product of the original graph with itself.

A method that exploits the shape similarity scores is proposed in [19]. This method uses an unsupervised clustering algorithm, aiming to capture the manifold structure of the image relations by defining a neighborhood for each data point in terms of a mutual k-nearest neighbor graph. The DOA (Distance Optimization Algorithm) is presented in [29]. 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.

Recently, *contextual information* has also been considered for improving the effectiveness of image retrieval [17, 49, 31, 36]. The objective of these methods is somehow mimic the human behavior on judging the similarity among objects by considering specific *contexts*. More specifically, the notion of *context* can refer to updating image similarity measures by taking into account information encoded on the ranked lists defined by a CBIR system [36]. Similarly to aproaches based on global ranking, these methods take information about relationships among images for re-ranking. In [31], the notion of context refers to the nearest neighbors of a query image. A similarity measure is proposed for assessing how similar two ranked lists are. An extension of this approach was proposed in [36]. A clustering method is used for representing the contextual information. In [31], a family of contextual measures of similarity between distributions is introduced. These contextual measures are then applied to the image retrieval problem as a re-ranking method.

The *Contextual Re-Ranking Algorithm* proposed in this paper aims to exploit contextual information for image re-ranking tasks. An important novelty of the Contextual Re-Ranking Algorithm consists in the use of image processing techniques for contextual information representation and processing. The proposed method is flexible in the sense it can be easily tailored to different CBIR tasks, considering shape, color and texture

descriptors. Furthermore, it also can be used for rank aggregation and for combining post-processing methods.

2.2 Rank Aggregation

Different CBIR descriptors produce different rankings. Further, it is intuitive that different descriptors may provide different but complementary information about images, and therefore their combination may improve ranking performance. An approach for improving CBIR systems consists in using *rank aggregation* techniques. Basically, rank aggregation approaches aim to combine different rankings in order to obtain a more accurate one.

Although rank aggregation problem has a long and interesting history that goes back at least two centuries [12, 24], it has been receiving great attention by the computer community in the last few decades. Rank aggregation is being employed in many new applications [24, 12], such as document filtering, spam webpage detection, meta-search, word association finding, multiple search, biological databases, and similarity search. Commonly, different rank aggregation approaches consider that objects highly ranked in many ranked lists are likely to be relevant [7]. For estimating the relevance of an object, given a ranked list, both *rank positions* [6] and *retrieval scores* [14] are considered.

Recently, learning to rank approaches are being considered [13]. Their objective is to use machine learning techniques to combine different CBIR descriptors. Rank aggregation also can be thought as an unsupervised regression, in which the goal is to find an aggregate ranking that minimizes the distance to each of the given ranked lists [37]. It also can be seen as the problem of finding a ranking of a set of elements that is "closest to" a given set of input rankings of the elements [11, 35, 12].

In general, by using supervised or unsupervised techniques, rank aggregation methods consider only scores or positions for producing new rankings. The rich contextual information encoded in relationships among images is ignored. In this paper, we exploit these relationships for rank aggregation.

Differently from the aforementioned methods, in our work, the similarity (in terms of effectiveness measures) between descriptors to be combined is considered for rank aggregation tasks. It is expected that uncorrelated systems would produce different rankings of the relevant objects, even when the overlap in the provided ranked lists is high. This observation is consistent with the statement that the combination with the lowest error occurs when the classifiers are independent and non-correlated [7].

3 Problem Definition

This section presents a formal definition for problems discussed in this paper. Section 3.1 presents a definion of the re-ranking problem considering contextual information. Section 3.2 presents a definition of the rank aggregation problem.

3.1 Re-Ranking Method

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

Let $\mathcal{D}$ be an *image descriptor* which can be defined [33] as a tuple (ϵ, ρ), where:

- $\epsilon\colon \hat{I} \to \mathbb{R}^n$ *is a function, which extracts a feature vector $v_{\hat{I}}$ from an image $\hat{I}$.*
- $\rho\colon \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ *is a distance function that computes the distance between two images according to 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 a query image img_q, we can compute a ranked list R_{img_q} in response to the posed query by taking into account the distance matrix A. The ranked list $R_{img_q} = \{img_i, img_j, \dots, img_N\}$ can be defined as a permutation of the collection $\mathcal{C}$, such that, if img_i is ranked higher than img_j, then $\rho(img_q, img_i) < \rho(img_q, img_j)$. We can also take each image $img_i \in \mathcal{C}$ as a query image img_q, in order to obtain a set $\mathcal{R} = \{R_{img_1}, R_{img_2}, \dots, R_{img_N}\}$ of ranked lists for each image img_i ($1 \le i \le N$) of collection $\mathcal{C}$. A re-ranking method that considers relations among all images in a collection can be represented by function f_r, such that f_r takes as input the distance matrix A and the set of ranked lists $\mathcal{R}$ for computing a new distance matrix $\hat{A}$:

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

Based on the distance matrix $\hat{A}$, collection images can be re-ranked, that is, a new set of ranked lists can be obtained. The Contextual Re-Ranking Algorithm, detailed in Section 4.2, consists in a implemention of function f_r.

3.2 Rank Aggregation Method

Let $\mathcal{C}$ be an image collection and let $\mathcal{D} = \{D_1, D_2, \dots, D_m\}$ be a set of m image descriptors. The set of descrip-

tors $\mathcal{D}$ can be used for computing a set of distances matrices $\mathcal{A} = \{A_1, A_2, \dots, A_m\}$. As discussed in previous subsection, for each distance matrix $A_i \in \mathcal{A}$, a set of ranked lists $\mathcal{R}_i = \{R_1, R_2, \dots, R_N\}$ can be computed. Let $\mathcal{R}_{\mathcal{A}} = \{\mathcal{R}_1, \mathcal{R}_2, \dots, \mathcal{R}_m\}$ be a set of sets of ranked lists (one set $\mathcal{R}_i$ for each matrix $\mathcal{A}_i$), the objective of rank aggregation methods that consider relationships among images is to use the sets $\mathcal{A}$ and $\mathcal{R}_{\mathcal{A}}$ as input for computing a new distance matrix $\hat{A}_c$:

$$\hat{A}_c = f_a(\mathcal{A}, \mathcal{R}_{\mathcal{A}}) \tag{2}$$

Based on the combined distance matrix $\hat{A}_c$, a new set of ranked lists can be computed. The Contextual Rank Aggregation Algorithm, detailed in Section 4.3, consists in a implemention of function f_a.

4 Contextual Methods

This section presents our methods for image re-ranking and rank aggregation considering contextual information. Section 4.1 discusses the contextual information representation used by our methods. Section 4.2 presents the re-ranking algorithm while Section 4.3 presents the rank aggregation algorithm. Finally, Section 4.4 discusses the use of our approach for combining re-ranking methods.

4.1 Contextual Information Representation

Let $\mathcal{C}$ be an image collection and let $\mathcal{D}$ be an image descriptor, the distance function ρ defined by $\mathcal{D}$ can be used for computing the distance $\rho(img_i, img_j)$ among all images $img_i, img_j \in \mathcal{C}$ in order to obtain an $N \times N$ distance matrix A.

Our goal is to represent the distance matrix A as a gray scale image and analyze this image for extracting *contextual information* using image processing techniques. For the gray scale image representation, referenced in this paper as *context image* $\hat{I}$, we consider two reference images $img_i, img_j \in \mathcal{C}$.

Let the *context image* $\hat{I}$ be a gray scale image defined by the pair (D_I, f), where D_I is a finite set of pixels (points in $\mathbb{N}^2$, defined by a pair (x, y)) and $f\colon D_I \to \mathbb{R}$ is a function that assigns to each pixel $p \in D_I$ a real number. We define the values of f function in terms of the distance function ρ (encoded into matrix A) and reference images $img_i, img_j \in \mathcal{C}$.

Let $R_i = \{img_{i_1}, img_{i_2}, \dots, img_{i_N}\}$ be the ranked list defined by matrix A considering the reference image img_i as query image; and $R_j = \{img_{j_1}, img_{j_2}, \dots, img_{j_N}\}$ the ranked list of reference image img_j. In this
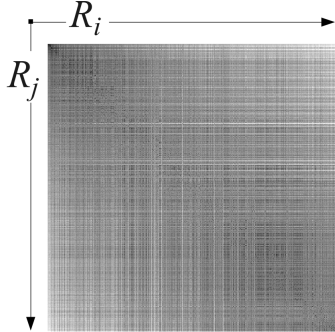
Fig. 1 Similar reference images.

Fig. 2 Non-similar reference images.

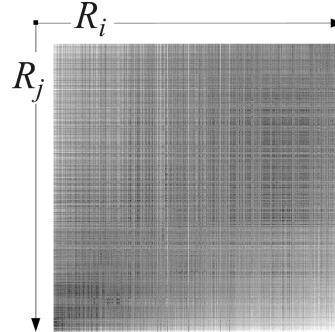**Fig. 3** Context image for similar reference images.

Fig. 4 Context image for non-similar reference images.

way, the axis of *context image* $\hat{I}$ are ordered according to the order defined by ranked lists R_i and R_j. Let $img_{i_x} \in R_i$ be an image at x position of ranked list R_i and $img_{j_y} \in R_j$ an image at y position of the ranked list R_j, the value of $f(x, y)$ (function that defines the gray scale of pixel $p(x, y)$) is defined as follows: $f(x, y) = \bar{\rho}(img_{i_x}, img_{j_y})$, where $\bar{\rho}$ is defined by the distance function ρ normalized in the interval [0,255].

An example, considering two similar reference images (from MPEG-7 database [21]), is illustrated in Figure 1. The respective gray scale image representing matrix A is illustrated in Figure 3. An analogous example for non-similar images is showed in Figures 2 and 4.

The *context images* can represent a great source of information about an image collection and distance among images. A single context image contains information about all distances among images and their spatial relationship defined by the ranked lists of the reference images. In other words, a single pixel can relate four collection images: the two reference images (that defines the position of the pixel, according to their ranked lists) and the two images whose distance defines the grayscale value of the pixel. Another important advantage for this image representation of distance among images is the possibility of using a large number of image processing techniques.

In this paper, our goal is to exploit useful *contextual information* provided by context images. Low distance values (similar images) are associated with dark pixels in the image, while high values (non-similar images) refers to non-black pixels. Considering two similar images as reference images, the beginning of two ranked lists should have similar images as well. This behavior creates a *dark region* in top left corner of a context image (as we can observe in Figure 3). This region

represents a neighborhood of similar images with low distances.

The top left corner represents images at the first position of the ranked lists of the two reference images, whose accuracy is higher than any other region in context image. We aim to characterize *contextual information* by analyzing this region using image processing techniques. These information will be used by the re-ranking method presented in next section.
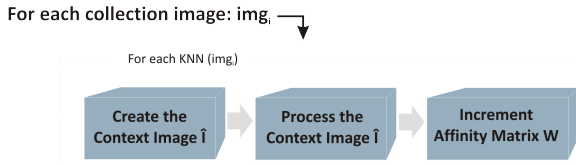
Other regions of context images could also be of interest. Considering similar reference images, the region close to the main diagonal, for example, contains more dark pixels (low distances) than the remaining of the image. Once the ranked lists of reference images are similar, pixels close to the main diagonal represent distances between similar images. The use of other regions of context images in image re-ranking tasks is left as future work.

4.2 The Contextual Re-Ranking Algorithm

Given an image $img_i \in \mathcal{C}$, we aim to process *contextual information* of img_i by constructing *context images* for each one of its *k-nearest neighbors* (based on distance matrix A). We use an affinity matrix W to store the results of processing *contextual information*. Let N be the size of collection $\mathcal{C}$, the affinity matrix W is an $N \times N$ matrix where $W[k, l]$ represents the similarity between images img_k and img_l.

We apply image processing techniques to process the *context images* that consider img_i and each one of its *k-nearest neighbor* to update the affinity matrix W. The same process is performed for all $img_i \in \mathcal{C}$. Since all images of $\mathcal{C}$ are processed, the affinity matrix W

Perform along T iterations:

For each collection image: img$_i$

For each KNN (img$_i$)

Create the Context Image Î → Process the Context Image Î → Increment Affinity Matrix W

Compute Distance Matrix A$_{t+1}$ *(from Affinity Matrix W)*

Perform Re-Ranking *(based on A$_{t+1}$)*

Fig. 5 The Contextual Re-Ranking Algorithm.

is used as input for computing a new distance matrix A_{t+1} (where t indicates the current iteration).

Based on the new distance matrix A_{t+1}, a new set of ranked lists are computed. These steps are repeated along several iterations. Finally, after a number T of iterations a re-ranking is performed based on final distance matrix A_T in order to obtain the final set of ranked lists. The main steps of Contextual Re-Ranking Algorithm are illustrated in Figure 5. Algorithm 1 outlines the complete re-ranking method, that is detailed in the following.

Algorithm 1 Contextual Re-Ranking Algorithm

Require: Original distance matrix A
Ensure: Processed distance matrix A_T
1: $t \leftarrow 0$
2: $A_t \leftarrow A$
3: **while** $t < T$ **do**
4:   $initializeAffinityMatrix(W, 1)$
5:   **for all** $img_i \in C$ **do**
6:     $k \leftarrow 1$
7:     **for all** $img_j \in KNN(img_i)$ **do**
8:       $ctxImg \leftarrow createContextImage(img_i, img_j, A_t, L)$
9:       $ctxImg' \leftarrow processContextImage(ctxImg, L)$
10:       $W \leftarrow updateAffinityMatrix(ctxImg', W, k)$
11:       $k \leftarrow k + 1$
12:     **end for**
13:   **end for**
14:   $A_{t+1} \leftarrow computeDistanceMatrix(W)$
15:   $performReRanking(A_{t+1})$
16:   $t = t + 1$
17: **end while**

The affinity matrix W is initialized with value 1 for all positions in step 4. *Context images* are created in step 7, as explained in Section 4.1, considering img_i (image being processed) and img_j (current neighbor of img_i) as reference images. The parameter L refers to the size of the square in the top left corner of *context image* that will be analyzed.

Image processing techniques are applied to *context images* in step 8. Our goal is to identify dense regions of dark pixels. Dark pixels indicate low distance values and, therefore, similar images. These regions represent the set of similar images at first positions of both ranked lists whose distances to each other are low. We use a threshold for obtaining a binary image and then identify dark pixels. The threshold l used is computed based on normalization given by average and maximum distance values contained in $L \times L$ square in top left corner of *context image*:

$$l = \frac{avg(\rho(img_p, img_q))}{max(\rho(img_p, img_q))} \tag{3}$$

with $p, q < L$.

In the following, we apply a median filter for determining regions of dense black pixels. The nonlinear median filter, often used for removing noise, is used in our approach aiming to correct distance among images. Basically, we consider that "*wrong*" distances can be considered and represented as "*noise*" and the median filter is used to filter this noise out. More specifically, consider a dense region of black pixels at the top left corner of a context image. It represents a set of similar images (low distances) at the top positions of ranked lists of reference images. Consider a white pixel in this region, indicating a high distance between two images. By taking into account the contextual information given by the region of the pixel (position and other close pixels), its very likely that the distance represented by this pixel is incorrect. In this scenario, the median filter replaces the white pixel by a black pixel. Similar reasoning can be applied to isolated black pixels in white regions. We should note that, in extreme situations, in which the CBIR descriptors completely confuse similar and non-similar images, there is less contextual information available in the context images.

Figure 6 illustrates an example of a binary image and Figure 7 shows the same image after applying the median filter (with a masks 3×3).
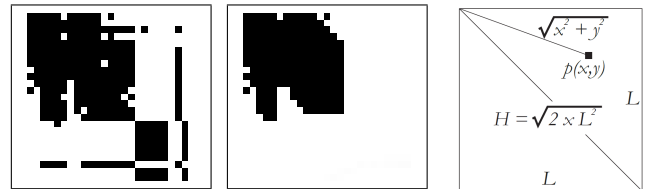
Fig. 6 Example of binary image.   **Fig. 7** Example of filtered image.   **Fig. 8** Updates of matrix W.

Step 9 updates the affinity matrix W based on the *context images*. For updating, only black pixels (and their positions) are considered. The objective is to give more relevance to pixels next to the origin $(0, 0)$, e.g., pixels that represent the beginning of ranked lists. The
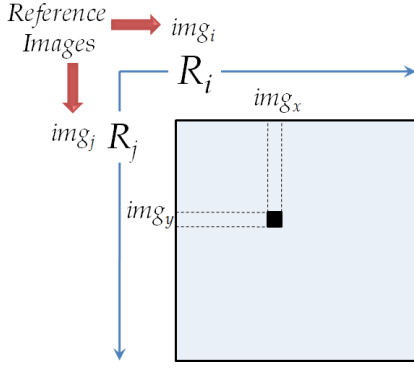
Fig. 9 Relationship among images provided by a single pixel of a context image.

importance of neighbors should also be considered: neighbors at first positions should be considered more relevant when updating W.

Let $img_i \in \mathcal{C}$ be the current image being processed. Let img_j be the k (such that $k < K$) neighbor of img_i. Let img_i and img_j be reference images and let $\hat{I}(D_I, f)$ be the *context image* after thresholding and applying the median filter. Let L be the size of the top left corner square that should be processed and let $p(x,y) \in D_I$ be a black pixel ($f(x,y) = 0$), such that $x, y < L$. The pixel $p(x,y)$ represents the distance between images img_x and img_y such that the image img_x is the image at the position x of the ranked list R_i and the image img_y is the image at position y of ranked list R_j.

Let $H = \sqrt{2 \times L^2}$ be the maximum distance of a pixel $p(x,y)$ to origin $(0,0)$, as illustrated in Figure 8. Let $W[x,y]$ represent the similarity between images img_{i_x} and img_{i_y}. Then, for each black pixel $p(x,y)$, the matrix W receives five updates: the most relevant one refers to the similarity between images img_x and img_y; two updates refers to the relationship between the reference image img_i with images img_x and img_y; and two updates refers to the relationships between reference image img_j and images img_x and img_y. Figure 9 illustrates the relationship among these images provided by each black pixel in the context image. The update of the similarity score between img_x and img_y (the most relevant one) is computed as follows:

$$W[x,y] \leftarrow W[x,y] + [(K-k) \times (H/\sqrt{x^2+y^2})] \quad (4)$$

Note that low values of k, x, y (the beginning of ranked lists) leads to high increments of W. Smaller increments occur when k has high values and $x, y = L$. In this case, the term $H/\sqrt{x^2+y^2}$ is equal to 1. The remaining four updates (relationship among reference images and images img_x, img_y) are computed as follows:

$$W[i,x] \leftarrow W[i,x] + \frac{[(K-k) \times (H/\sqrt{x^2+y^2})]}{4} \quad (5)$$

$$W[i,y] \leftarrow W[i,y] + \frac{[(K-k) \times (H/\sqrt{x^2+y^2})]}{4} \quad (6)$$

$$W[j,x] \leftarrow W[j,x] + \frac{[(K-k) \times (H/\sqrt{x^2+y^2})]}{4} \quad (7)$$

$$W[j,y] \leftarrow W[j,y] + \frac{[(K-k) \times (H/\sqrt{x^2+y^2})]}{4} \quad (8)$$

Observe that these four updates have together the same weight of the first update. They are computed based on the position of img_x and img_y (x,y) in the ranked the lists of other images (reference images img_i, img_j), while the first update is given by a pixel that represents the distance between images img_x and img_y.

When all images have been processed, and therefore an iteration has finished, the affinity matrix W presents high values for similar images. But there may be positions of W that was not updated (e.g., in the case of non-similar reference images), and have the initial value 1. The new distance matrix A_{t+1} (step 12 of Algorithm 1) is computed as follows:

$$A_{t+1}[x,y] = \begin{cases} 1 + \bar{A}_t[x,y], & \text{if } W[x,y] = 1 \\ 2 \times (1/W[x,y]), & \text{if } W[x,y] > 1 \end{cases} \quad (9)$$

where $\bar{A}_t$ is the distance matrix A_t normalized in the interval $[0,1]$. When $W[x,y] = 1$, i.e., $W[x,y]$ was not updated by Equation 4, we use the old distance matrix A_t for determining values of A_{t+1}. Otherwise (when $W[x,y] > 1$), values of new distance matrix A_{t+1} is equal to the inverse of affinity matrix W. Since the smallest increment for W is 1 (and therefore $W[x,y] = 2$), the largest value of a new distance in A_{t+1} is 0.5. Therefore, we normalize the new distance values in the interval $[0,1]$ by multiplying distances by 2. A_{t+1} will have values in the interval $[0,2]$: (i) in the interval $[0,1]$, if $W[x,y] > 1$, and (ii) in the interval $[1,2]$, if $W[x,y] = 1$. A last operation is performed on the new distance matrix A_{t+1} for ensuring the simmetry of distances between images ($\rho(x,y) = \rho(y,x)$):

$$A_{t+1}[x,y] \leftarrow A_{t+1}[y,x] \leftarrow min(A_{t+1}[x,y], A_{t+1}[y,x]) \quad (10)$$

Finally, a re-ranking is performed based on values of A_{t+1} (step 15 of Algorithm 1). At the end of T iterations, a new computed distance matrix A_T and a set of new ranked list are obtained.

4.3 The Contextual Rank Aggregation Algorithm

The presented re-ranking algorithm can be easily tailored to rank aggregation tasks. In this section, we present the *Contextual Rank Aggregation Algorithm*, aiming to combine the results of different descriptors. The main idea consists in applying the same iterative approach based on context images, but using the affinity matrix W for accumulating updates of different descriptors at the first iteration.

Algorithm 2 outlines the rank aggregation algorithm. We can observe that the agorithm is very similar to the re-ranking algorithm (Algoithm 1). It also considers an iterative approach and the context images for the contextual information processing. Note that the main difference relies on lines 8-13 of Algorithm 2, that are executed only at the first iteration, when different matrices $A_d \in \mathcal{A}$ of different descriptors are being combined.

Algorithm 2 Contextual Rank Aggregation Algorithm

Require: Set of distance matrices $\mathcal{A}$
Ensure: Processed distance matrix A_T
 1: $t \leftarrow 1$
 2: **while** $t < T$ **do**
 3:    $initializeAffinityMatrix(W, 1)$
 4:    **for all** $img_i \in C$ **do**
 5:      **for all** $img_j \in KNN(img_i)$ **do**
 6:        $k \leftarrow 1$
 7:        **if** $t = 1$ **then**
 8:          **for all** $A_d \in \mathcal{A}$ **do**
 9:            $ctxImg \leftarrow$ $createContextImage(img_i, img_j, A_d, L)$
10:            $ctxImg' \leftarrow processContextImage(ctxImg, L)$
11:            $W \leftarrow updateAffinityMatrix(ctxImg', W, k)$
12:            $k \leftarrow k + 1$
13:          **end for**
14:        **else**
15:          $ctxImg \leftarrow createContextImage(img_i, img_j, A_t, L)$
16:          $ctxImg' \leftarrow processContextImage(ctxImg, L)$
17:          $W \leftarrow updateAffinityMatrix(ctxImg', W, k)$
18:          $k \leftarrow k + 1$
19:        **end if**
20:      **end for**
21:    **end for**
22:    $A_{t+1} \leftarrow computeDistanceMatrix(W)$
23:    $performReRanking(A_{t+1})$
24:    $t = t + 1$
25: **end while**

4.4 Combining Post-Processing Methods

We defined a generic re-ranking algorithm as a implementation of a function $f_r(A, \mathcal{R})$ in Section 3.1. Both the input and output of the function f_r is given by a distance matrix (since the set of ranked lists $\mathcal{R}$ can be computed based on distance matrix).

In this way, a matrix obtained from another post processing method (other implementation of f_r) can be submitted to our re-ranking algorithm. Different approaches may exploit different relationships among images and further improve the effectiveness of CBIR systems. Contextual information can be exploited by our re-ranking algorithm even after other methods have already been processed. We present experiments for validating this conjecture in Section 5.4.

5 Experimental Evaluation

In this section, we present the set of conducted experiments for demonstrating the effectiveness of our method. We analyzed and compared our method under several aspects. In Section 5.1.1, we present an analysis of the Contextual Algorithm considering: the impact of parameters and image processing techniques on the re-ranking algorithm; and a brief discussion about complexity and efficiency.

In Section 5.2, we discuss the experimental results for our re-ranking method. Section 5.2.1 presents results of the use of our method for several shape descriptors, considering the well-known MPEG-7 database [21]. Sections 5.2.2 and 5.2.3 aim to validate the hypothesis that our method can be applied to general image retrieval tasks. In addition to shape descriptors, we conducted experiments with color and texture descriptors.

Section 5.3 presents experimental results of our method on rank aggregation tasks. Section 5.4 presents experimental results of our re-ranking method combined with other post-processing methods. Finally, we also conducted experiments aiming to compare our results to state-of-the-art related post-processing and rank aggregation methods in Section 5.5.

All experiments were conducted considering all images in the collections as query images. Results presented in the paper (MAP and Recall@40 scores) represent an average result.

5.1 Experiment 1: Analysis of Contextual Re-Ranking Algorithm

In this section, we evaluated the Contextual Re-Ranking Algorithm with regard to different aspects. Section 5.1.1 analyzes the impact of parameters in effectiveness results. Section 5.1.2 evaluates the relevance of image processing techniques for the algorithm. Section 5.1.3 discusses aspects of efficiency and computational complexity.
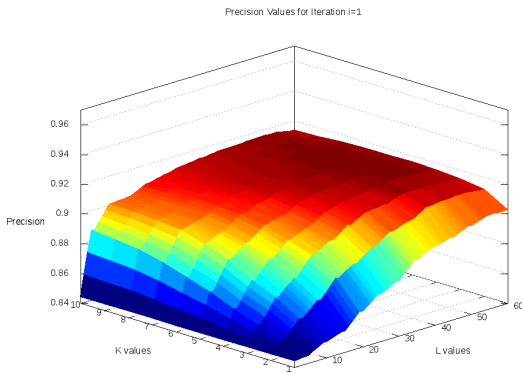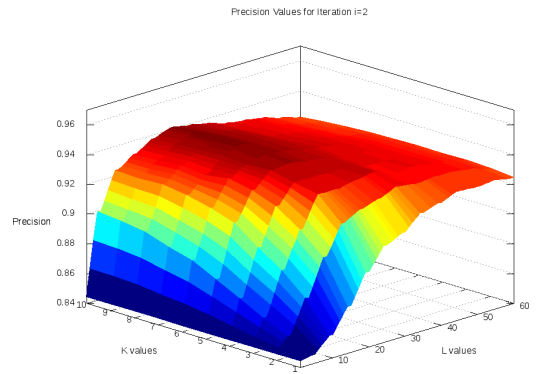
Fig. 10 Iteration 1: 91.84%, K=5, L=50.

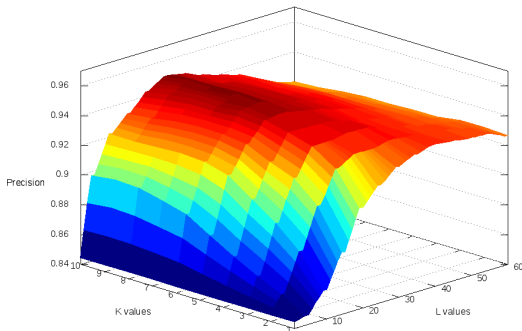**Fig. 11** Iteration 2: 94.41%, K=7, L=25.
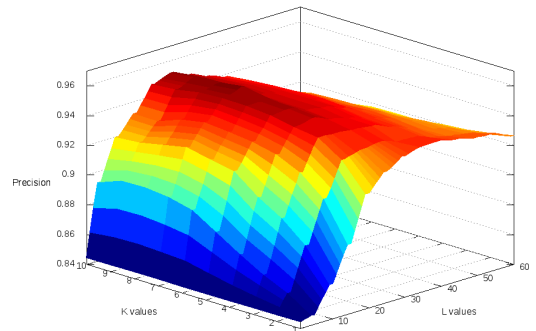

Fig. 12 Iteration 3: 95.29%, K=7, L=25.

Fig. 13 Iteration 4: 95.66%, K=7, L=25.

5.1.1 Impact of Parameters

The execution of Algorithms 1 and 2 considers three parameters: *(i)* K - number of neighbors used as reference images; *(ii)* L - size of top left square of context image to be analyzed; and *(ii)* T - number of iterations that the algorithm is executed.

To evaluate the influence of different parameter settings on the retrieval scores and for determining the best parameters values we conducted a set of experiments. We use the MPEG-7 database [21] with the so-called bullseye score (*Recall@40*), which counts all matching objects within the 40 most similar candidates. The MPEG-7 data set consists of 1400 silhouette images grouped into 70 classes. Each class has 20 different shapes. 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 [28] shape descriptor.

Retrieval scores are computed ranging parameters K in the interval [1,10] and L in the interval [1,60] (with increments of 5) for each iteration. Figures 10, 11, 12, and 13 show surfaces that represent retrieval scores for iterations 1, 2, 3, and 4 respectively. For each iteration, the best retrieval score was determined.

We observed that the best retrieval scores increased along iterations and parameters converged for values
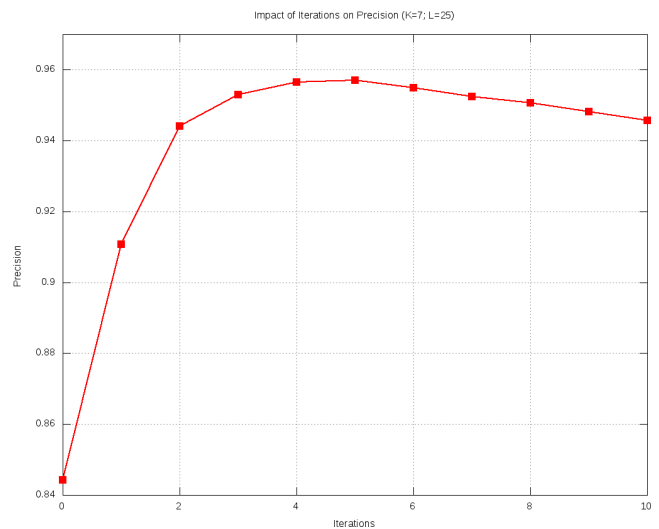

Fig. 14 Impact of iterations on precision.

$K = 7$ and $L = 25$. Figure 14 illustrates the evolution of precision according to the iterations of re-ranking algorithm. The best retrieval score was reached at iteration $T = 5$: **95.71%**. Note that these parameters may change for databases with very different sizes. The parameter values $K = 7$, $L = 25$, and $T = 5$ was used for all experiments, except for Soccer color dataset (described in Scetion 5.2.3). Since this dataset is very smaller than others, we used $K = 3$.

5.1.2 Impact of Image Processing Techiniques

In this section we aim to evaluate the impact of the image processing techniques in effectiveness results. For the experiments, we consider the MPEG-7 [21] dataset (with Recall@40 score), the CFD [28] shape descriptor and the parameters values defined in Section 5.1.1. We evaluated the method with regard to the follows aspects:

– **Median filter:** we have disabled the median filter (considering only the thresholding). The retrieval score obtained was 93.94%.
– **Thresholding:** we have disabled the thresholding and filter steps (considering updating for all pixels in context images). The effectiviness result obtained was 92.89%.
– **Masks of median filter:** we evaluated the effectivenes of the method for different sizes of masks (3, 5, and 7). We have obtained for masks 3, 5, and 7, respectively **95.71%**, 95.36%, and 95.18%.

The experimental results demostrate the positive impact of image processing techniques in effetiveness results of Contextual Re-Ranking Algorithm. The best retrieval score was obtained when a thresholding and a median filter of mask 3×3 are used.

5.1.3 Aspects of Effiency

This paper focuses on the presentation of Contextual Re-Ranking algorithm and its effectiveness evaluation. The focus on effectiveness is justified by the fact that the execution of the algorithm is expected to be off-line, as in other post-processing methods [43]. This subsection aims to briefly discuss some aspects of efficiency and computational complexity.

Let $\mathcal{C}$ be an image collection with N images. The number of context images that should be processed is equal to $(N \times K \times T)$. The size of context images that impacts the number of updates in matrix W is given by L^2 pixels. Since the parameters K, T, and L have fixed values independent of N, the asymptotic computional complexity of mains steps of the algorithm (image processing and W matrix updating steps) is $O(N)$. Other stpes of the algorithm have different complexities. The matrices A and W are recomputed ($O(N^2)$) at each iteration. The re-ranking step computes a sort operation ($O(NlogN)$) for all images ($O(N^2logN)$). However, these steps admit optimizations: once the updatings for matrix W impacts a small subset of positions (depending on the size L^2 of context image), the matrices do not require to be totally recomputed and the ranked lists do not require to be totally sorted again.

The Contextual Re-Ranking algorithm can also be massively parallelized, since there is no dependece between processing of different context images at a same iteration. Optimizations and parallelization issues will be investigated in future work.

Note also that other post-processing methods use matrices multiplication approaches [47, 48] and graph algorithms [43], both with complexity of $O(N^3)$.

We evaluated the computation time of Contextual Re-Ranking algorithm for MPEG-7 dataset ($N = 1400$), using the parameters defined in Section 5.1.1 ($K = 7$, $L = 25$ and $T = 5$), executing in a Linux PC Core 2 Quad and using a C implementation. This execution took approximately 6s.

5.2 Experiment 2: Re-Ranking

In this section, we present a set of conducted experiments for demonstrating the effectiveness of our method. Various post-processing methods [19, 48, 47, 26] have been evaluated considering only one type of visual property (usually, either color or shape). We aim to evaluate the use of our method in a general way for several CBIR tasks. We compared results for several descriptors (shape, color, and texture) in differents datasets. The measure adopted is *Mean Average Precision (MAP)*, geometrically referred as the average area below Precision $\times$ Recall curves considering different queries. Table 1 presents results for 11 image descriptors in 3 different datasets. As we can observe in Table 1, the Contextual Re-Ranking method presents positive effectiveness gains for all descriptors (including shape, color, and texture). The gains ranged from +1.37% to +18.90%, with 8.57% on the average. We conducted a paired t-test and conclude that there is a 99.9% chance of difference between the means (before and after the re-ranking) being statistical significantly. Next subsections present the descriptors and datasets used for shape, color and texture experiments.
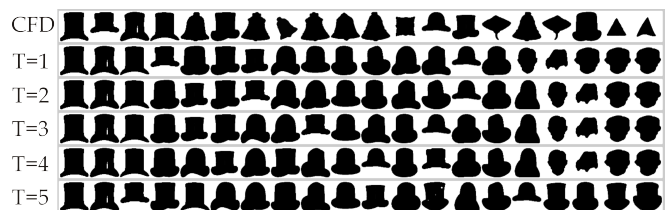
5.2.1 Shape Descriptors

We evaluate the use of our method with five shape descriptors considering the MPEG-7 database [21]: Beam Angle Statistics (BAS) [2], Segment Saliences (SS) [34], Inner Distance Shape Context (IDSC) [22], Contour Features Descriptor (CFD) [28], and Aspect Shape Context (ASC) [23]. 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 +5.29% to +16.80%, with 10.56% on average. Figure 15 presents the percentage gain obtained by Contextual

Table 1 Contextual Re-Ranking Evaluation on Content-Based Image Retrieval Tasks.

Descriptor	Type	Dataset	Score (MAP)	Contextual Re-Ranking	Gain
SS [34]	Shape	MPEG-7	37.67%	**44.79%**	+18.90%
BAS [2]	Shape	MPEG-7	71.52%	**76.60%**	+7.10%
IDSC [22]	Shape	MPEG-7	81.70%	**87.39%**	+6.96%
ASC [23]	Shape	MPEG-7	85.28%	**89.82%**	+5.32%
CFD [28]	Shape	MPEG-7	80.71%	**92.76%**	+14.93%
AIR [15]	Shape	MPEG-7	89.39%	**94.49%**	+5.71%
GCH [39]	Color	Soccer	32.24%	**33.02%**	+2.42%
ACC [16]	Color	Soccer	37.23%	**39.86%**	+7.06%
BIC [38]	Color	Soccer	39.26%	**43.04%**	+9.63%
LBP [25]	Texture	Brodatz	48.40%	**49.06%**	+1.37%
CCOM [20]	Texture	Brodatz	57.57%	**63.67%**	+10.60%
LAS [40]	Texture	Brodatz	75.15%	**78.48%**	+4.43%

Table 2 Contextual Re-Ranking for Shape Descriptors on MPEG-7 *(Recall@40)*.

Shape Descriptor	Score [%]	Contextual Re-Ranking	Gain
SS [34]	43.99%	**51.38%**	+16.80%
BAS [2]	75.20%	**82.43%**	+9.61%
IDSC [22]	85.40%	**91.84%**	+7.54%
ASC [23]	88.39%	**93.07%**	+5.29%
CFD [28]	84.43%	**95.71%**	+13.36%
AIR [15]	93.67%	**99.80%**	+6.54%

Fig. 16 Evolution of rankings according to iterations on MPEG-7 [21] database (first column contains the query image): the first row presents the results of CFD [28] shape descriptor; the remaining rows present the results of the *Contextual Re-Ranking Algorithm* for each iteration.

the iterations. The first row presents 20 results for a query image (first column) according to the CFD [28] shape descriptor. The remaining rows present the results for each iteration of Contextual Re-Ranking algorithm. We can observe the significant improvement in terms of precision, ranging from 40% (on the ranking computed by the CFD [28] descriptor) to 100% at the fifth iteration of the re-ranking algorithm.

5.2.2 Texture Descriptors

The experiments considered three texture descriptors: Local Binary Patterns (LBP) [25], Local Activity Spectrum (LAS) [40], and Color Co-Occurrence Matrix (CCOM) [20]. We used the Brodatz [5] dataset, a popular dataset for texture descriptors evaluation. The Brodatz dataset are composed of 111 different textures. Each texture is divided into 16 blocks, such that 1776 images are considered. Our re-ranking method presents positive gains, presented in Table 1, ranging from +1.37% to 10.60%.

5.2.3 Color Descriptors

Three color descriptors was considered for our evaluation: Auto Color Correlograms (ACC) [16], Border

Fig. 15 Contextual re-ranking percent gain for CFD [28] shape descriptor on MPEG-7 classes.

Re-Ranking algorithm for CFD [28] descriptor considering each of 70 shape classes in MPEG-7 dataset. Note that bullseye score was improved over 30% for several classes.

The iterative behavior of the Contextual Re-Ranking algorithm can be observed in results illustrated in Figure 16. The figure shows the evolution of rankings along

Table 3 Contextual rank aggregation on several content-based image retrieval tasks (*Mean Average Precision*)

| Image Descriptors	Type	Dataset	Score (MAP)
CFD [28]	Shape	MPEG-7	80.71%
ASC [23]	Shape	MPEG-7	85.28%
CFD [28] + ASC [23]	**Shape**	**MPEG-7**	**98.77%**
ACC [16]	Color	Soccer	37.23%
BIC [38]	Color	Soccer	39.26%
ACC [16] + BIC [38]	**Color**	**Soccer**	**42.14%**
CCOM [20]	Texture	Brodatz	63.67%
LAS [40]	Texture	Brodatz	75.15%
CCOM [20] + LAS [40]	**Texture**	**Brodatz**	**81.63%**

/ Interior Pixel Classification (BIC) [38], and Global Color Histogram (GCH) [39]. The experiments were conducted on a database used in [44] and composed by images from 7 soccer teams, containing 40 images per class. We can observe positive gains for all color descriptors, presented in Table 1, ranging from +2.42% to 9.63%.

5.3 Experiment 3: Rank Aggregation

This section aims to evaluate the use of our re-ranking method to combine different CBIR descriptors. We selected two descriptors for each visual property (shape, color, and texture): descriptors with best effectiveness results are selected (except for MPEG-7 dataset, where AIR [15] descriptor is very close to the maximum scores). Table 3 presents the MAP scores obtained for rank aggregation considering these descriptors. We can observe significant gains compared with each isolated descriptors results. Figure 17 illustrates the Precision x Recall curves of shape descriptors CFD [28] and ASC [23] in different situations: before and after applying the Contextual Re-Ranking algorithm, and after using it for rank aggregation. As it can be observed, for both re-ranking and rank aggregation, very significant gains in terms of precision have been achieved.

5.4 Experiment 4: Combining Post-Processing Methods

In this section, we aim to evaluate the use of our re-ranking method combined with other post-processing methods. We considered two post-processing approaches: Distance Optimization Algorithm (DOA) [28] and Mutual kNN Graph [19]. Table 4 presents the results for *MAP* and *Recall@40* measures. The gains are positives, ranging from +0.30% to +0.90%.
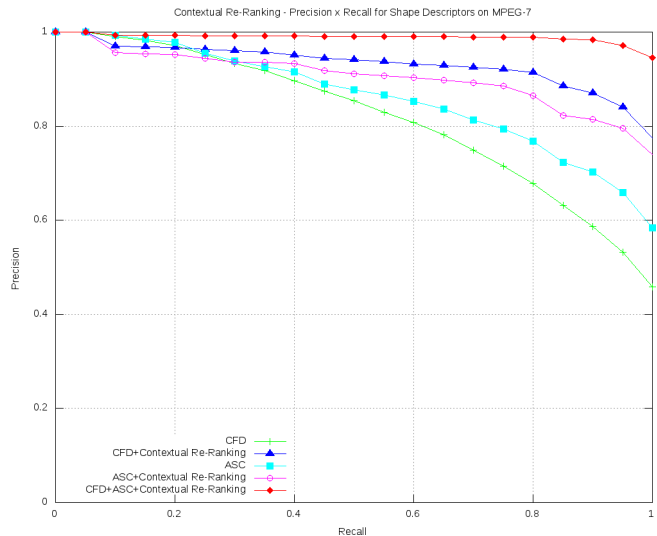
Fig. 17 Contextual re-ranking and rank aggregation for shape descriptors.

Table 4 Combining post-processing methods using contextual re-ranking on MPEG-7 database (Recall@40).

Algorithm	Score	Contextual Re-Ranking	Gain
DOA [28]	92.56%	**93.39%**	**+0.90%**
Mutual kNN Graph [19]	93.40%	**93.68%**	**+0.30%**

5.5 Experiment 5: Comparison to Other Approaches

We also evaluated our method in comparison with other state-of-the-art post-processing methods. We used the MPEG-7 database with the bullseye score again. Table 5 presents results of our Contextual Re-Ranking algorithm and several other post-processing methods in different tasks (re-ranking, rank aggregation, and combining post-processing methods). We also present the retrieval scores for some descriptors that has been used as input for these methods. We can observe that the Contextual Re-Ranking method presents high effectiveness scores when compared to state-ot-the-art approaches. Note that our method has the best effectiveness performance when compared to all other post-processing methods in rank aggregation tasks.

6 Conclusions

In this work, we have presented a new re-ranking method based on contextual information. The main idea consists in creating gray scale image representations of distance matrix and performing a re-ranking based on information extracted from these images. We conducted a large set of experiments, considering several descriptors and datasets. Experimental results demonstrate the use

Table 5 Post-processing methods comparison on MPEG-7 database *(Recall@40)*.

Algorithm	Shape Descriptor	Score	Gain
Shape Descriptors			
Data Driven Generative Models (DDGM) [42]	-	80.03%	-
Contour Features Descriptor (CFD) [28]	-	84.43%	-
Inner Distance Shape Context (IDSC) [22]	-	85.40%	-
Shape Context (SC) [4]	-	86.80%	-
Aspect Shape Context (ASC) [23]	-	88.39%	-
Articulation-Invariant Representation (AIR) [15]	-	93.67%	-
Post-Processing Methods			
Graph Transduction (LP) [47]	IDSC [22]	91.00%	+6.56%
Contextual Re-Ranking	**IDSC [22]**	**91.84%**	**+7.54%**
Distance Optimization Algorithm (DOA) [28]	CFD [28]	92.56%	+9.63%
Contextual Re-Ranking	**ASC [23]**	**93.07%**	**+5.29%**
Locally Constrained Diffusion Process [48]	IDSC [22]	93.32%	+9.27%
Mutual kNN Graph [19]	IDSC [22]	93.40%	+9.37%
Locally Constrained Diffusion Process [48]	IDSC [22]+St. I [41]	93.80%	+9.84%
Locally Constrained Diffusion Process [48]	IDSC [22]+St. I [41]	94.85%	+11.07%
Locally Constrained Diffusion Process [48]	IDSC [22]+St. I&II [41]	95.60%	+11.94%
Contextual Re-Ranking	**CFD [28]**	**95.71%**	**+13.36%**
Locally Constrained Diffusion Process [48]	ASC [23]	95.96%	+8.56%
Contextual Re-Ranking	**AIR [15]**	**99.80%**	**+6.54%**
Tensor Product Graph [46]	AIR [15]	99.99%	+6.75%
Combining Post-Processing Methods			
Contextual Re-Ranking + DOA [28]	**CFD [28]**	**93.39%**	**+10.61%**
Contextual Re-Ranking + kNN Graph [19]	**IDSC [22]**	**93.68%**	**+9.70%**
Rank Aggregation Methods			
Co-Transduction [3]	IDSC [22]+DDGM [42]	97.31%	-
Co-Transduction [3]	SC [4]+DDGM [42]	97.45%	-
Co-Transduction [3]	SC [4]+IDSC [22]	97.72%	-
Contextual Re-Ranking	**CFD [28]+IDSC [22]**	**98.95%**	-
Contextual Re-Ranking	**CFD [28]+ASC [23]**	**99.38%**	-

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 MPEG-7 dataset.

Future work focuses on: *(i)* parallelizing and optimizing the proposed algorithm; *(ii)* using other image processing techniques, as dynamic thresholding and other filtering approaches; *(iii)* analyzing other regions of context images; *(iv)* investigating the use of context images for other applications (for clustering and computing the similarity between ranked lists, for example).

7 Acknowledgment

References

1. Abowd GD, Dey AK, Brown PJ, Davies N, Smith M, Steggles P (1999) Towards a better understanding of context and context-awareness. In: Proceedings of the 1st international symposium on Handheld and Ubiquitous Computing, HUC '99, pp 304–307

2. Arica N, Vural FTY (2003) Bas: a perceptual shape descriptor based on the beam angle statistics. Pattern Recogn Lett 24(9-10):1627–1639

3. Bai X, Wang B, Wang X, Liu W, Tu Z (2010) Co-transduction for shape retrieval. ECCV 3:328 – 341

4. Belongie S, Malik J, Puzicha J (2002) Shape matching and object recognition using shape contexts. PAMI 24(4):509–522

5. Brodatz P (1966) Textures: A Photographic Album for Artists and Designers. Dover

6. Coppersmith D, Fleischer LK, Rurda A (2010) Ordering by weighted number of wins gives a good ranking for weighted tournaments. ACM Trans Algorithms 6:55:1–55:13

7. Croft WB (2002) Combining approaches to information retrieval. In: Croft WB, Croft WB (eds) Advances in Information Retrieval, The Information Retrieval, vol 7, Springer US, pp 1–36

8. Dai HJ, Lai PT, Tsai RTH, Hsu WL (2010) Global ranking via data fusion. In: Proceedings of the 23rd International Conference on Computational Linguistics: Posters, COLING '10, pp 223–231

9. Datta R, Joshi D, Li J, Wang JZ (2008) Image retrieval: Ideas, influences, and trends of the new age. ACM Comput Surv 40:5:1–5:60

10. Diaz F (2005) Regularizing ad hoc retrieval scores. In: CIKM '05, pp 672–679

11. Dwork C, Kumar R, Naor M, Sivakumar D (2001) Rank aggregation methods for the web. In: Proceedings of the 10th international conference on World Wide Web, ACM, New York, NY, USA, WWW '01, pp 613–622, DOI http://doi.acm.org/10.1145/371920.372165, URL http://doi.acm.org/10.1145/371920.372165

12. Fagin R, Kumar R, Mahdian M, Sivakumar D, Vee E (2004) Comparing and aggregating rankings with ties. In: Proceedings of the twenty-third ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems, PODS '04, pp 47–58

13. Faria FF, Veloso A, Almeida HM, Valle E, da S Torres R, Gonçalves MA, Jr WM (2010) Learning to rank for content-based image retrieval. In: MIR '10, pp 285–294

14. Fox EA, Shaw JA (1994) Combination of multiple searches. In: The Second Text REtrieval Conference (TREC-2), NIST, NIST Special Publication, vol 500-215, pp 243–252

15. Gopalan R, Turaga P, Chellappa R (2010) Articulation-invariant representation of non-planar shapes. In: Proceedings of the 11th European conference on computer vision conference on Computer vision: Part III, ECCV'10, pp 286–299

16. Huang J, Kumar SR, Mitra M, Zhu WJ, Zabih R (1997) Image indexing using color correlograms. In: CVPR '97, p 762

17. Jégou H, Harzallah H, Schmid C (2007) A contextual dissimilarity measure for accurate and efficient image search. In: CVPR, pp 1–8

18. Ji S, Zhou K, Liao C, Zheng Z, Xue GR, Chapelle O, Sun G, Zha H (2009) Global ranking by exploiting user clicks. In: SIGIR '09, pp 35–42

19. Kontschieder Peter BH Donoser Michael (2009) Beyond pairwise shape similarity analysis. ACCV '09 pp 655–666

20. Kovalev V, Volmer S (1998) Color co-occurence descriptors for querying-by-example. In: MMM '98, p 32

21. Latecki LJ, Lakmper R, Eckhardt U (2000) Shape descriptors for non-rigid shapes with a single closed contour. In: CVPR, pp 424–429

22. Ling H, Jacobs DW (2007) Shape classification using the inner-distance. PAMI 29(2):286–299, DOI http://dx.doi.org/10.1109/TPAMI.2007.41

23. Ling H, Yang X, Latecki LJ (2010) Balancing deformability and discriminability for shape matching. ECCV 3:411 – 424

24. Liu YT, Liu TY, Qin T, Ma ZM, Li H (2007) Supervised rank aggregation. In: WWW 2007, pp 481 – 490

25. Ojala T, Pietikäinen M, Mäenpää T (2002) Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. PAMI 24(7):971–987

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

27. Pedronette DCG, da S Torres R (2010) Exploiting contextual information for image re-ranking. CIARP 1:541–548

28. Pedronette DCG, da S Torres R (2010) Shape retrieval using contour features and distance optmization. In: VISAPP, vol 1, pp 197 – 202

29. Pedronette DCG, da S Torres R (2011) Exploiting clustering approaches for image re-ranking. Journal of Visual Languages and Computing 22(6):453–466

30. Pedronette DCG, da S Torres R (2011) Exploiting contextual information for rank aggregation. ICIP

31. Perronnin F, Liu Y, Renders JM (2009) A family of contextual measures of similarity between distributions with application to image retrieval. CVPR pp 2358–2365

32. Qin T, Liu TY, Zhang XD, Wang DS, Li H (2008) Global ranking using continuous conditional random fields. In: NIPS, pp 1281–1288

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

34. da S Torres R, Falcão AX (2007) Contour Salience Descriptors for Effective Image Retrieval and Analysis. Image and Vision Computing 25(1):3–13

35. Schalekamp F, Zuylen A (1998) Rank aggregation: Together were strong. In: In Proc. of 11th ALENEX, pp 38–51

36. Schwander O, Nielsen F (2010) Reranking with contextual dissimilarity measures from representational bregmanl k-means. In: VISAPP, vol 1, pp 118–122

37. Sculley D (2007) Rank aggregation for similar items. In: SDM

38. Stehling RO, Nascimento MA, Falcão AX (2002) A compact and efficient image retrieval approach based on border/interior pixel classification. In:

CIKM '02, pp 102–109

39. Swain MJ, Ballard DH (1991) Color indexing. IJCV 7(1):11–32

40. Tao B, Dickinson BW (2000) Texture recognition and image retrieval using gradient indexing. JVCIR 11(3):327 – 342

41. Temlyakov A, Munsell BC, Waggoner JW, , Wang S (2010) Two perceptually motivated strategies for shape classification. CVPR 1:2289 – 2296

42. Tu Z, Yuille AL (2004) Shape matching and recognition - using generative models and informative features. ECCV pp 195–209

43. Wang J, Li Y, Bai X, Zhang Y, Wang C, Tang N (2011) Learning context-sensitive similarity by shortest path propagation. Pattern Recogn 44:2367–2374

44. van de Weijer J, Schmid C (2006) Coloring local feature extraction. In: ECCV, Springer, vol Part II, pp 334–348

45. Yang L, Ji D, Zhou G, Nie Y, Xiao G (2006) Document re-ranking using cluster validation and label propagation. In: CIKM '06, pp 690–697

46. Yang X, Latecki LJ (2011) Affinity learning on a tensor product graph with applications to shape and image retrieval. In: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, pp 2369 –2376

47. Yang X, Bai X, Latecki LJ, Tu Z (2008) Improving shape retrieval by learning graph transduction. ECCV 5305:788–801

48. Yang X, Koknar-Tezel S, Latecki LJ (2009) Locally constrained diffusion process on locally densified distance spaces with applications to shape retrieval. In: CVPR, pp 357–364

49. Zhao D, Lin Z, Tang X (2007) Contextual distance for data perception. In: ICCV

50. Zhu X (2005) Semi-supervised learning with graphs. PhD thesis, Pittsburgh, PA, USA, chair-Lafferty, John and Chair-Rosenfeld, Ronald