Distances Correlation for Re-Ranking in Content-Based Image Retrieval

Daniel Carlos Guimarães Pedronette and Ricardo da S. Torres
RECOD Lab - Institute of Computing - IC
University of Campinas - UNICAMP
Campinas, Brazil
dcarlos@ic.unicamp.br, rtorres@ic.unicamp.br

***Abstract*—Content-based image retrieval relies on the use of efficient and effective image descriptors. One of the most important components of an image descriptor is concerned with the distance function used to measure how similar two images are. This paper presents a clustering approach based on distances correlation for computing the similarity among images. Conducted experiments involving shape, color, and texture descriptors demonstrate the effectiveness of our method.**

***Keywords*-content-based image retrieval; re-ranking; distance optimization; clustering;**

I. Introduction

Technological improvements in image acquisition and the decreasing cost of storage devices have enabled the dissemination of large image collections. In this scenario, there is the need of methods for indexing and retrieving these data. One of the most common approaches to support image searches relies on the use of Content-Based Image Retrieval (CBIR) systems.

Basically, given a query image, a CBIR system aims at retrieving the most similar images in a collection by taking into account image visual properties (such as, shape, color, and texture). Collection images are ranked in decreasing order of similarity, according to a given *image descriptor*. A descriptor is characterized by [1]: *(i)* an extraction algorithm to encode image features into feature vectors; and *(ii)* a similarity measure to compare two images. The similarity between two images is computed as a function of the distance of their feature vector.

Several efforts have been proposed for improving effectiveness of CBIR approaches. Example of recent initiatives include the use of new image descriptors and matching algorithms [2], [3]. However, in general, these approaches perform only pairwise image analysis, that is, they compute similarity (or distance) measures considering only pair of images, ignoring the rich information encoded in the relations among several images.

Some post-processing methods have been proposed for improving effectiveness of information retrieval tasks [4]–[10]. Some efforts were put on post-processing the similarity scores by analyzing the relations among all documents in a given collection. The objective is to exploit these relations aiming at increasing the effectiveness of retrieval tasks [4]–[8]. Promising results have been reported [4]–[8] on applying various post-processing methods in CBIR applications.

Some approaches try to infer the similarity between two images by analyzing the similarity between their ranked lists [7]. However, these methods have not been designed to exploit the similarity of ranked lists encoded in distances correlation measures. In this paper, we present a new post-processing algorithm that exploits the concept of *distances correlation* for re-ranking images. This paper also illustrates the use of this algorithm in several CBIR tasks considering color, texture, and shape properties.

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

II. Related Work

A re-ranking method is proposed in [8]. Images are analyzed using hierarchical agglomerative clustering and the image ranks are adjusted according to the distance of clusters with regard to the query image. The approach is evaluated using images retrieved by taking into account color histogram information.

The influence among shape similarities in an image collection is analized in [5]. Markov chains are used to perform a diffusion process on a graph formed by a set of shapes, where the influences of other shapes are propagated. The approach introduces a locally constrained diffusion process and a method for densifying the shape space by adding synthetic points.

A method that exploits the shape similarity scores is proposed in [4]. This method uses a 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. Experiments that demonstrate improvements in shape retrieval results and clustering are reported.

A graph transduction learning approach is introduced in [6]. The algorithm computes the shape similarity of a pair of shapes in the context of other shapes as opposed to considering only pairwise relations. This method is an application of semi-supervised label propagation algorithm [11]. The objective of this algorithm is, based on some labeled data, to propagate labels in a weighted graph defined by a similarity measures between elements of a database. The label propagation algorithm has also been used for re-ranking documents in information retrieval applications [9]. The approach proposed in [6] extends label propagation algorithm on an unsupervisioned way, using queries as initial labels. The label propagation process is interrupted when a certain number of iterations is reached.

A distance optimization algorithm has been proposed in [7]. This algorithm clusters shapes by taking into account the similariy among ranked lists. Distances between shapes are updated based on created clusters aiming at improving the retrieval effectiveness. In this paper, we extend the distance optimization algorithm (described in more details in the next section) by exploiting the concept of distances correlation to create clusters and re-rank images.

III. The Distance Optimization Algorithm

Let $\mathcal{C}$=$\{img_1, img_2, \ldots, img_N\}$ be an image collection and $\mathcal{D}$ an image descriptor. We can use $\mathcal{D}$ to obtain an $N \times N$ distance matrix A that describes the pairwise relations between all shapes in terms of a distance measure. Based on matrix A, we can compute a set of ranked lists $\mathcal{R}$=$\{R_1$,R_2,$\ldots$,$R_N\}$ for each image $img_i \in \mathcal{C}$. For each R_i, images are ranked according to their distance to img_i.

The distance optimization algorithm proposed in [7] aims at improving the retrieval rate of a given descriptor $\mathcal{D}$ for a image collection $\mathcal{C}$ based on the set of ranked lists $\mathcal{R}$. The algorithm explores the fact that if two images are similar, their ranked lists should be similar as well. Basically, the algorithm redefines the distance among images, given the similarity of their ranked lists. A clustering approach is used for that. The algorithm performs two main steps in an iterative way until a convergence criterion is reached:

1) Create clusters, by exploring information of ranked lists;
2) Update (decrease) distances among images of a same cluster;

Images are assigned to the same cluster if they have similar ranked lists. Next, distances among images belonging to same cluster are updated (decreased). Redefining distances implicates on performing a re-ranking of the set $\mathcal{R}$. This process is repeated until the "quality" of the formed clusters does not improve and, therefore, "good" ranked lists are created. A *cohesion* measure is proposed for evaluating the clusters and and is used as a convergence crterion. Algorithm 1 presents the distance optimization method.

Algorithm 1 Distance Optimization Algorithm [7]

```
Require: Distance matrix $W$
Ensure: Optimized distance matrix $W_o$
1: $lastCohesion \leftarrow 0$
2: $currentCohesion \leftarrow computeCohesion(W)$
3: while $curCohesion > lastCohesion$ do
4:    $Cls \leftarrow createClusters(W)$
5:    W ← updateDistances(W,Cls)
6:    $lastCohesion \leftarrow currentCohesion$
7:    $currentCohesion \leftarrow computeCohesion(W)$
8: end while
9: $W_o \leftarrow W$
```

The most relevant steps of Algorithm 1 reside on steps 4 and 5. In these steps, the algorithm creates clusters and updates distances among images based on these clusters. The remaining steps compute and verify the convergence criterion (*cohesion* measure). The method for distances updating (step 5) will be the focus of this paper.

Next sections present how steps 4 and 5 were implemented in [7]. This implementation will be used as baseline in our experiments. Section IV presents a new approach to implement the distances updating method. Basically, the proposed method exploits distances correlation information to perform an adaptative updating for distances among images.

A. Graph-based Clustering

The clustering method proposed in [7] assigns two images to the same cluster if they are considered *cluster-similar*. This concept is defined formally below:

Let $G(V, E)$ be a directed and weighted graph, where a vertex $v \in V$ represents an image. The weight w_e of edge $e = (v_i, v_j) \in E$ is defined by the ranking position of image img_j (v_j) at the ranked list of img_i (v_i).

Definition 1. *Let (k, l) be a ordered pair. Two images img_i and img_j are (k, l)-**similar** if $w_{e_{i,j}} \leq k$ and $w_{e_{j,i}} \leq l$, where $e_{i,j} = (i_i, i_j)$ is the edge between images img_i and img_j.*

Definition 2. *Let $S_p = \{(k_0, l_0), (k_1, l_1), \ldots, (k_m, l_m)\}$ be a set of ordered pairs. Two images, img_i and img_j, are **cluster-similar** according to S_p, if $\exists (k_a, l_a) \in S_p | img_i$ and img_j are (k_a, l_a)-similar.*

Figure 1 presents an example of a set of ordered pairs that can be used to define a cluster-similar function based on the ranked lists similarities.

B. Distances Updating

Once created clusters, the distance optimization algorithm updates the distances among all database images. The method for updating distances is perfomed as follows: as we

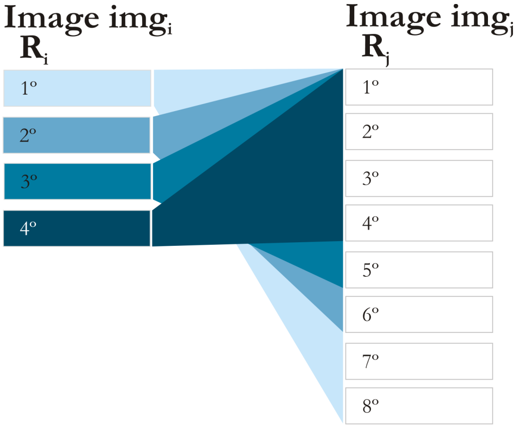

Figure 1. Example of a set S_p of ordered pairs used in a cluster-similar function. In this example, $S_p = \{(1,8),(2,6),(3,5),(4,4)\}$ [7].

discussed previously, considering each image of collection $\mathcal{C}$ as query q_i, we can generate a ranked list R_i, based on the distances computed by a *descriptor*. Therefore, we can compute a set of ranked lists $\mathcal{R}$=$\{R_1$,R_2,…,$R_N\}$, one for each database image.

Let R_i=$\{img_1$,img_2,…,$img_N\}$ be a ranked list associated to a query image q_i. For each image $img_j \in R_i$, we check to which cluster img_j was assigned. If image img_j belongs to the same cluster of the query image q_i, the distance between img_j and q_i is decreased, (multiplied by a constant $\lambda < 1$). These new distance values are used to re-rank collection images.

IV. New Re-Ranking Approach

Although very effective, the distance optimization algorithm presents possibilities for extensions. Once created the clusters, the updating is perfomed using only the λ constant value to compute the new distance value, ignoring any other information encoded in the relations among images. This paper presents a new approach to update the distances between images in an adaptative way, considering distances correlation information.

A. Bidimensional Space Between Pairwise Objects Distance

Consider the image space $\mathbb{R}^2$ defined by the image collection $\mathcal{C}$=$\{img_1, img_2, \ldots, img_N\}$ and a distance function $\rho : \mathcal{C} \times \mathcal{C} \rightarrow \mathbb{R}$, where $\mathbb{R}$ denotes real numbers. Consider $\rho(i,j) \geq 0$ for all (i,j) and $\rho(i,j) = 0$ if $i = j$.

We can use this space for analyzing the similarity of collection images with regard to two arbitrary images $img_i, img_j \in \mathcal{C}$ (these images are used as reference). Consider a graphic representation of the image collection $\mathcal{C}$ on a Cartesian coordinate system. Let $img_l \in \mathcal{C}$ be an image, we can plot a point representing img_l on the plane, considering its distances to the images img_i and img_j:

Given two reference images img_i and img_j, we can consider a plane where the *x axis* represents the distances of collection images with regard to image img_i and the *y axis* represents the distances of collection images with regard to img_j. The position of an image $img_l \in \mathcal{C}$ is given by the ordered pair ($\rho(i,l)$,$\rho(j,l)$), where $\rho(i,l)$ and $\rho(j,l)$ are the distances of img_l to the reference images img_i and img_j respectively. We can use this same approach to determine the position of all collection images.

Figure 2 shows the graphic representation of an image collection (MPEG-7 dataset [12]) by taking into account two reference images that are very similar. In this example we have used the CFD descriptor [7] to compute distance among images. Note that the distribution of images follows a linear behavior.

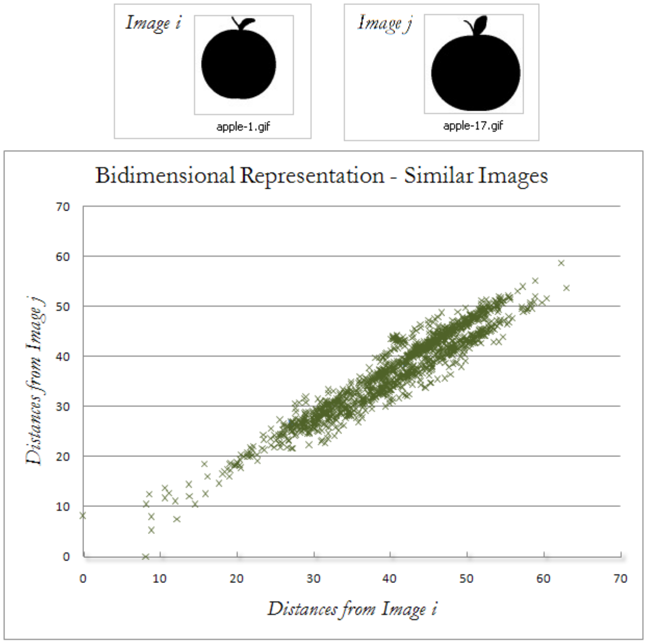

Figure 2. Bidimensional space representation for similar images.

Figure 3 shows the same representation now considering two reference images that are not similar, according to the same descriptor. Note that the two graphic representations present very distinct characteristics. Our goal is to use this information for image re-ranking.

Besides the graphic representation we can use statistical measures to characterize the images distribution represented in Figures 2 and 3. Our goal is to measure the similarity between images img_i and img_j using distances from images img_i and img_j to other images. In statistics, a measure of association is a numerical index which describes the strength or magnitude of a relationship among variables [13]. In this paper we analyze this relationship by using Pearson's Correlation Coefficient:

$$r = \frac{\sum_{i=1}^{n}(X_i - \overline{X})(Y_i - \overline{Y})}{\sqrt{\sum_{i=1}^{n}(X_i - \overline{X})^2}\sqrt{\sum_{i=1}^{n}(Y_i - \overline{Y})^2}}$$

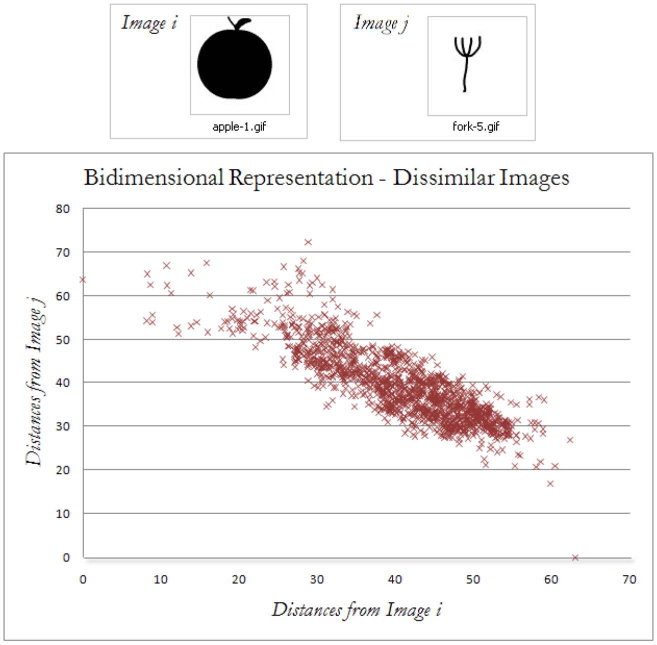

Figure 3. Bidimensional space representation for non-similar images.

Pearson's correlation coefficient r for continuous data ranges from -1 to +1, where $r = -1$ data lie on a perfect straight line with a negative slope; $r = 1$ data lie on a perfect straight line with a positive slope.

Given the correlation measure r, we can use this information to update distances on an adaptative way. This new proposed approach will be presented in the next section.

As discussed in [6], if the database is large, the computation of post-processing methods with all N objects may become impractical. A solution is proposed considering only the distances of K nearest neighbors given in the ranked lists of reference images img_i and img_j. Note that we consider $KNNs$ of image img_i and $KNNs$ of image img_j for composition of vectors X and Y (used for Pearson correlation computation). Thus, the size of these vectors may range from K (when $KNNs$ of img_i and img_j have the same elements) to $2 \times K$ (when all elements of $KNNs$ of img_i and img_j are different).

B. Distances Update based on Correlation

As mentioned in Section III, the distance optimization algorithm considers only clusters and a $\lambda < 1$ constant for updating distances among images. This is due to the fact that the *cluster-similar* function provides only binary information.

However, this problem does not affect the correlation measure r. Therefore, we propose to update distance among images by using correlation information.

Our strategy considers the set of ranked lists $\mathcal{R}$=$\{R_1$,R_2,$\ldots$,$R_N\}$. Let $R_i \in \mathcal{R}$ be the ranked list produced by matrix A for and image img_i. Let Cl_i be the cluster to which image img_i was assigned. The update approach is performed by dividing the ranked list R_i in three segments as follows:

- Seg_1: an image $img_j \in Seg_1$, if $img_j \in Cl_i$, i.e., if images img_i and img_j belong to the same cluster Cl_i;
- Seg_2: an image $img_k \in c \cdot KNN$ of R_i and $k \notin Seg_1$, i.e., if the index i_k of the image k in ranked list R_i is such that $i_k < c \cdot K$ and img_k does not belong to the same cluster of img_i;
- Seg_3: an image $l \notin Seg_1$ and $l \notin Seg_2$.

Figure 4 illustrates the three segments of a given ranked list R_i according to these criteria. For each segment of the ranked list, a different update method is performed.

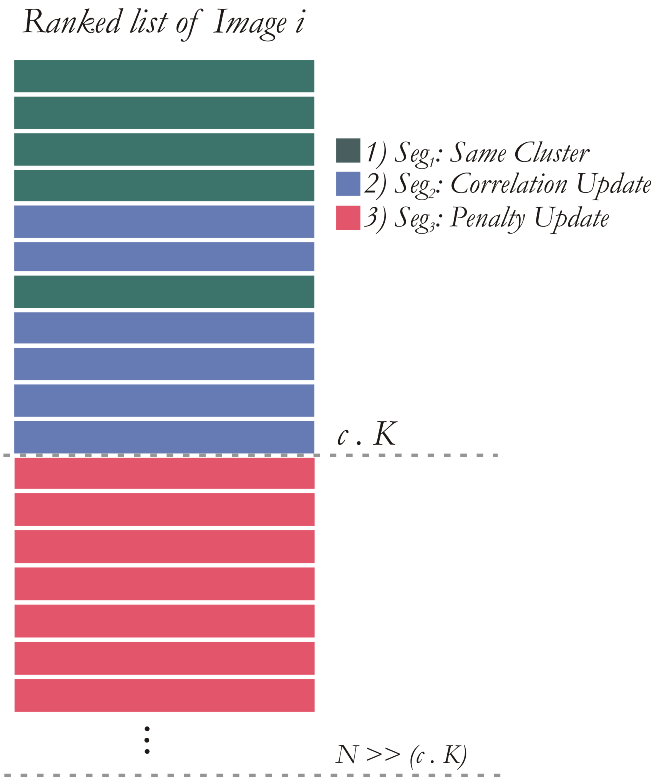

Figure 4. Segmentation of ranked lists in the new distance update approach.

Let $\rho(i,j)$ be the current distance between the images i and j, and $\widehat{\rho}(i,j)$ the distance after the update respectively, the value of updated distance is computed for each segment as follows:

- Seg_1: $\widehat{\rho}(i,j) = \rho(i,j) \cdot \lambda$
- Seg_2: $\widehat{\rho}(i,j) = \rho(i,j) \cdot (1 + [(1-\lambda) \cdot (1-\overline{r})])$
- Seg_3: $\widehat{\rho}(i,j) = \rho(i,j) \cdot [1 + (1-\lambda)]$

The $\overline{r}$ value represents the value of correlation normalized in the interval [0,1].

The central idea behind this approach is to explore the correlation information for updating distances. When images

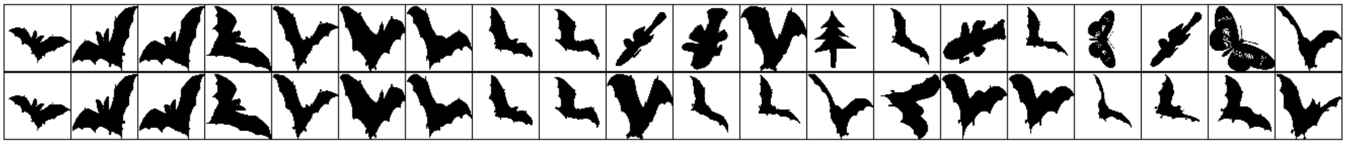

Figure 5. First row: retrieval results for the CFD [7] shape descriptor (first image as a query). Second row: retrieval results for the same shape descriptor after distance optimization.

are in set Seg_1 (same cluster) the distance are multiplied by constant $\lambda < 1$, as initially proposed in distance optimization algorithm [7]. However, when images are in set Seg_2 (fuzzy region on ranked list), an adaptative update is performed: the value for multiplying distance ranges in the interval $[1, 1 + (1 - \lambda)]$, according to the correlation between images. The remain images (the set Seg_3) are multiplied by a fixed value greater than 1: $1+(1-\lambda)$, which we name as *penalty update*.

Note that images in the same cluster have their distances reduced. Otherwise, all remaining images in the ranked list have its distances increased. The reasons for only images in the Seg_2 set suffer influence of correlation are the same of previous discussed for the choice of KNN images for correlation computation: the computation of correlation for updating distances for all images may become impractical for large databases. Remind that Seg_2 considers only $c \cdot K$ neighbors of image img_i.

V. Evaluation

In this section, we present a set of conducted experiments for demonstrating the applicability and effectiveness of our method. We analyzed and compared our method under several aspects.

In Section V-A, we present an analysis of the effects of correlation on the distance optimization algorithm.

Section V-B presents results of applying our method for several shape descriptors, considering the well-known MPEG-7 database [12]. We also conducted experiments comparing our results to the original distance optimization algorithm and to state-of-the-art related post-processing methods.

Finally, Sections V-C, V-D, and V-E aims 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.

A. Experimental Analysis - Correlation Impact

In this section we aim to analyze the effects of the use of distances correlation. Our goal is to verify these effects both on distances computed by a given descriptor (and its ranked lists), and on the behavior of distance optimization algorithm. Figure 5 illustrates an example of results comparison of CFD shape descriptor [7] before and after the application of distance optimization using correlation.

We analyzed the impact of using correlation information on updating distances and how this use affects the distance optimization algorithm. Besides retrieval results (presented in next section) the convergence of algorithm is a good indicator of this behavior.

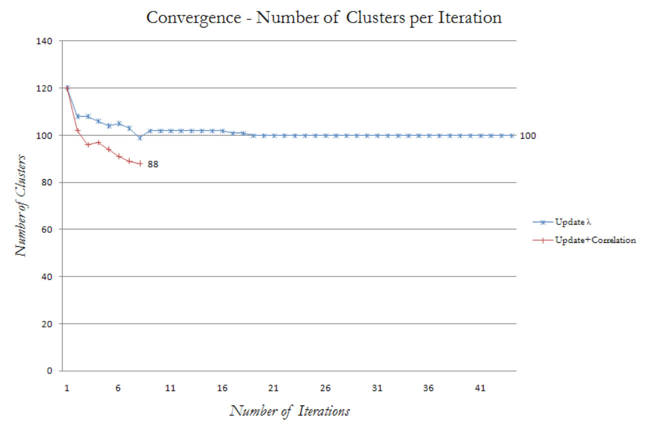

Figure 6. Convergence: number of cluster per iteration.

Figure 6 illustrates the evolution of distance optimization algorithm in terms of number of clusters by iteration. We consider original algorithm and new distance update approach based on correlation. As we can observe, the correlation affects the algorithm by decreasing the number of cluster (next to expected number of clusters - 70) and the number of iterations necessary to reach this value.

The last performed analysis aims to verify the impact of distance optimization algorithm on distances and their correlations. For this analysis we construct the same graphic of Figure 3 (where an image collection is represented in terms of distances of two non-similar images) after the execution of distance algorithm based on correlation. This graphic is presented in Figure 7. We can observe very distinct sets of points: (i) points next to x axis (images similar to image i); (ii) points next to y axis (images similar to image j); (iii) central points (remaining images).

B. Experimental Results for Shape Descriptors

In this section we aim to evaluate our method with regard to three different aspects: *(i)* comparing the correlation approach with original distance optimization algorithm; *(ii)* comparing distance optimization algorithm with correlation approach to other related post-processing methods and *(iii)*

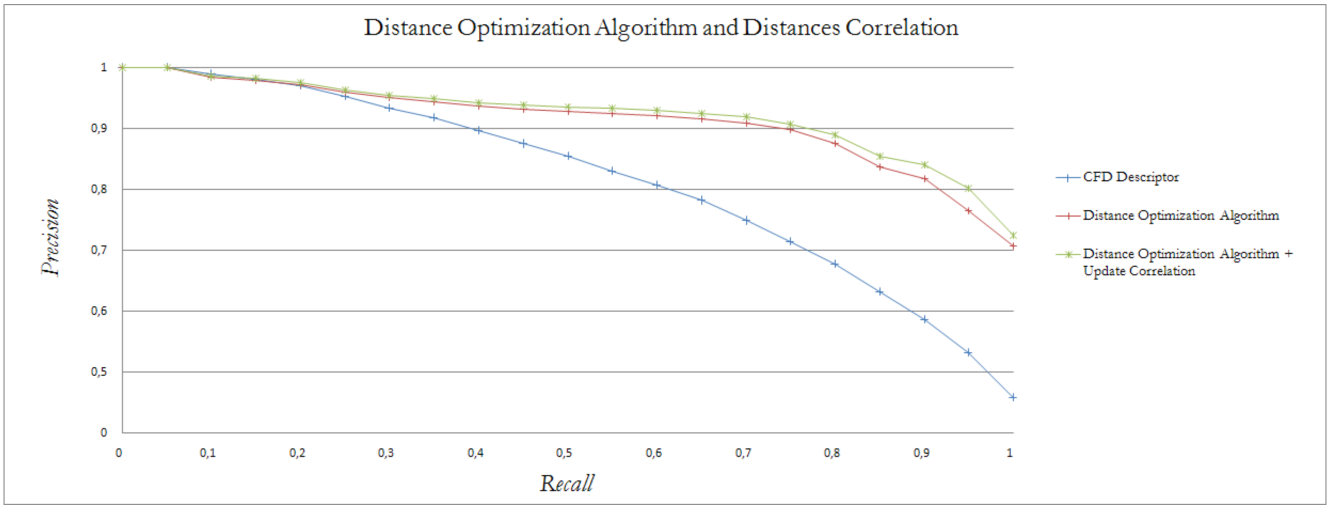

Figure 8. Precision vs. Recall: comparing results of distance optmization algorithm variations

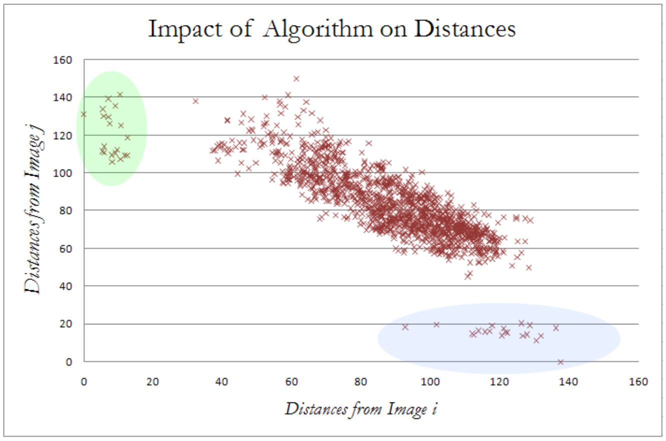

Figure 7. Impact of Algorithm on Distances.

Table I
POST-PROCESSING METHODS COMPARISON ON MPEG-7 DATABASE

Algorithm	**Descriptor**	**Score**	**Gain**
CFD [7]	-	84.43%	-
IDSC+DP [14]	-	85.40%	-
Graph Transduction [6]	IDSC+DP	91.00%	+6.56%
Distance Optmization [7]	CFD	92.56%	+9.63%
Constrained Diffusion Process [5]	IDSC+DP	93.32%	+9.27%
Mutual kNN Graph [4]	IDSC+DP	93.40%	+9.37%
DistOpt+UpCor	**CFD**	**93.62%**	**+10.88%**

Table II
DISTANCE OPTIMIZATION + CORRELATION FOR SHAPE DESCRIPTORS

Shape Descriptor	**Score**	**DistOpt+UpCor**	**Gain**
SS [15]	43.99%	50.93%	+15.78%
BAS [16]	75.20%	85.11%	+13.18%
IDSC+DP [14]	85.40%	90.02%	+5.40%
CFD [7]	84.43%	93.62%	+10.88%

evaluate the use of our method with several shape descriptors.

For the experiments we use the following parameters values: $\lambda = 0.95$, $K = 20$, and $c = 6$. For other parameters of distance optimization algorithm, we use the same values used in [7].

We compare the proposed correlation approach to original distance optimization algorithm on MPEG-7 database. Figure 8 presents the Precision vs. Recall curves for both methods. As we can observe, the approach that uses correlation presents better precision values.

We also evaluate our method in comparison to other state-of-the-art post-processing methods. We use MPEG-7 database with the so-called bullseye score, which counts all matching objects within the 40 most similar candidates. Since each class consists of 20 objects, the retrieved score is normalized with the highest possible number of hits (20 x 1400). In Table I, we present results of our method based on correlation (in bold), original distance optimization algorithm and three post-processing methods. We also present the IDSC+DP [14] descriptor, that has been used as input for these methods. Note that *Distance Optmization + Updating based on Correlation* has the best effectiveness performace when compared to all other post-processing methods.

Finally, we evaluate the use of our methods to other shape descriptors using the MPEG-7 database. Results are presented in Table II. Note that the effectiveness gains are always positive and ranges from +5.40% to +15.78%.

C. *Experimental Results for Texture Descriptors*

All discussed post-processing methods [4]–[8] have been evaluated their approaches for only one type of visual property (usually, either color or shape). Methods proposed in [4]–[7] used shape descriptors, while the method proposed

Table III
CORRELATION METHODS EVALUATION ON SEVERAL CONTENT-BASED IMAGE RETRIEVAL TASKS - MEAN AVERAGE PRECISION

Image Descriptor	**Type**	**Dataset**	**Score [%]** *(MAP)*	**Distance Optimization + Update Correlation**	**Gain**
SS [15]	Shape Descriptor	MPEG-7	37.67%	46.53%	+23.52%
BAS [16]	Shape Descriptor	MPEG-7	71.52%	81.05%	+13.32%
IDSC+DP [14]	Shape Descriptor	MPEG-7	81.70%	86.94%	+6.41%
CFD [7]	Shape Descriptor	MPEG-7	80.71%	91.79%	+13.73%
ACC [17]	Color Descriptor	Soccer Dataset	37.23%	42.46%	+14.05%
BIC [18]	Color Descriptor	Soccer Dataset	39.26%	38.16%	-2.80%
CCOM [19]	Texture Descriptor	Brodatz	57.57%	59.27%	+2.95%
LAS [20]	Texture Descriptor	Brodatz	75.15%	80.36%	+6.93%

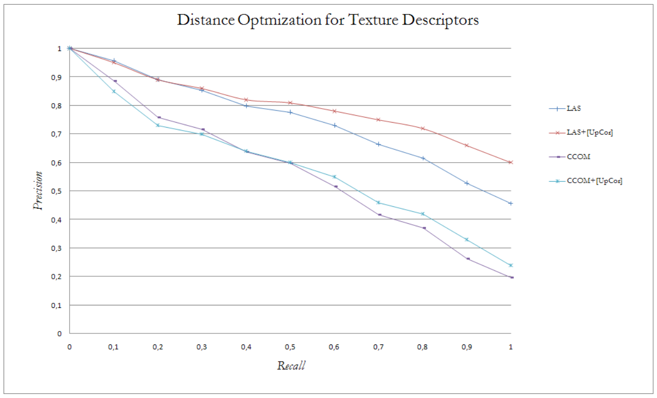

Figure 9. Distance Optmization + Correlation applied to Texture Descriptors.

in [8] used a color descriptor.

Our goal here is to evaluate the use of our method for several CBIR tasks (shape, color, and texture). In this section we aim to validate using image retrieval using texture descriptors. The comparasion uses two well-know texture descriptors and compared the effectiveness of retrieval before and after execution of distance optimization algorithm using correlation.

We used the Brodatz [21] dataset, a popular dataset for texture descriptors evaluation. Since the Brodatz dataset presents different categorization characteristics from MPEG-7 dataset, we changed same parameters of distance optimization algorithm [7]: we use $top_n = 30$, $th_{cohesion} = 55$, $top_{nToAdd} = 8$, and $K = 15$.

Figure 9 presents the precision vs. recall curve for descriptors CCOM [19] and LAS [20] and the curve after the execution of the distance optimization algorithm based on distances correlation approach. We can observe that, for LAS descriptor, the distance optimization algorithm improved approximately 15% on end of curve, without any loss of precision.

D. Experimental Results for Color Descriptors

We evaluate our method for two color descriptors: BIC [18] and ACC [17]. The experiments were conducted on a database used in [22] and composed by images from 7 soccer teams, containing 40 images per class.

The parameters of distance optimization algorithm were the same to the MPEG-7 database. Figure 10 presents the precision vs. recall curve for descriptors BIC [18] and ACC [17] before and after the use of distance optmization algorithm based on correlation. As we can see, for ACC descriptor there is a positive gain. The same behaviour was not observed for the BIC descriptor.

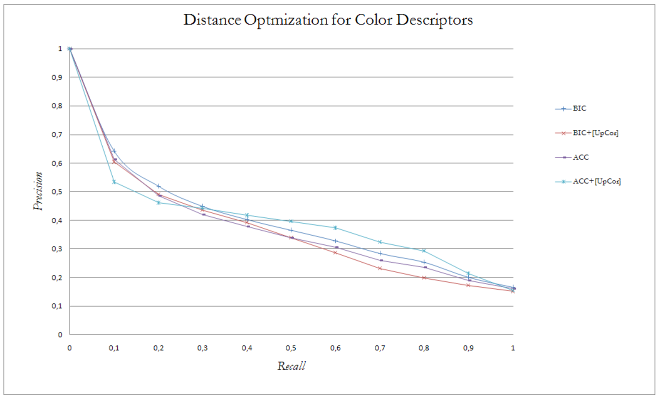

Figure 10. Distance Optmization + Correlation applied to Color Descriptors.

E. Experimental Results for General CBIR Tasks

Finally, we evaluate our method in a general way, comparing results for several descriptors (shape, color, texture) in differents datasets. The measure adopted is *Mean Average Precision (MAP)*, geometrically refered as the area below precision × recall curve.

Results are presented in Table III. Except for the BIC Color Descriptor, the correlation approach presents positive effectiveness gains for all descriptors, ranging from +2.95% to +23.52%.

VI. CONCLUSION

In this work, we presented the concept of *distances correlation* and proposed a new approach for a re-ranking method using this concept. The main idea consists in creating image clusters based on distances correlation and performing a reranking based on clusters and distances correlation information.

We conducted a large set of experiments and experimental results demonstrated the applicability of our method to 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.

Our future work will focus on using the proposed method in other information retrieval tasks. We also plan to investigate its use in relevance feedback strategies.

ACKNOWLEDGMENT

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

REFERENCES

[1] R. da S. Torres and A. X. Falcao, "Content-Based Image Retrieval: Theory and Applications," *Revista de Informática Teórica e Aplicada*, vol. 13, no. 2, pp. 161–185, 2006.

[2] P. F. Felzenszwalb and J. D. Schwartz, "Hierarchical matching of deformable shapes," *Computer Vision and Pattern Recognition (CVPR)*, pp. 1–8, 2007.

[3] L. Lin, X. Liu, and S.-C. Zhu, "Layered graph matching with composite cluster sampling," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 99, pp. 1426–1442, 2009.

[4] H. B. Peter Kontschieder, Michael Donoser, "Beyond pairwise shape similarity analysis," *Asian Conference on Computer Vision (ACCV)*, pp. 655–666, 2009.

[5] X. Yang, S. Koknar-Tezel, and L. J. Latecki, "Locally constrained diffusion process on locally densified distance spaces with applications to shape retrieval." in *Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2009, pp. 357–364.

[6] X. Yang, X. Bai, L. J. Latecki, and Z. Tu, "Improving shape retrieval by learning graph transduction," *European Conference on Computer Vision (ECCV)*, vol. 5305, pp. 788–801, 2008.

[7] D. C. G. Pedronette and R. da S. Torres, "Shape retrieval using contour features and distance optmization," in *International Conference on Computer Vision Theory and Applications (VISAPP)*, vol. 1, 2010, pp. 197 – 202.

[8] G. Park, Y. Baek, and H.-K. Lee, "Re-ranking algorithm using post-retrieval clustering for content-based image retrieval," *Information Processing and Management*, vol. 41, no. 2, pp. 177 – 194, 2005.

[9] L. Yang, D. Ji, G. Zhou, Y. Nie, and G. Xiao, "Document re-ranking using cluster validation and label propagation," in *CIKM '06: Proceedings of the 15th ACM international conference on Information and knowledge management*, 2006, pp. 690–697.

[10] X. Liu and W. B. Croft, "Cluster-based retrieval using language models," in *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, 2004, pp. 186–193.

[11] X. Zhu, "Semi-supervised learning with graphs," Ph.D. dissertation, Pittsburgh, PA, USA, 2005, chair-Lafferty, John and Chair-Rosenfeld, Ronald.

[12] L. J. Latecki, R. Lakmper, and U. Eckhardt, "Shape descriptors for non-rigid shapes with a single closed contour," in *Proc. IEEE Conference Computer Vision and Pattern Recognition*, 2000, pp. 424–429.

[13] H. Xiong, S. Shekhar, P.-N. Tan, and V. Kumar, "Exploiting a support-based upper bound of pearson's correlation coefficient for efficiently identifying strongly correlated pairs," in *KDD '04: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2004, pp. 334–343.

[14] H. Ling and D. W. Jacobs, "Shape classification using the inner-distance," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 2, pp. 286–299, 2007.

[15] R. da S. Torres and A. X. Falcao, "Contour Salience Descriptors for Effective Image Retrieval and Analysis," *Image and Vision Computing*, vol. 25, no. 1, pp. 3–13, January 2007.

[16] N. Arica and F. T. Y. Vural, "Bas: a perceptual shape descriptor based on the beam angle statistics," *Pattern Recogn. Lett.*, vol. 24, no. 9-10, pp. 1627–1639, 2003.

[17] J. Huang, S. R. Kumar, M. Mitra, W.-J. Zhu, and R. Zabih, "Image indexing using color correlograms," in *CVPR '97: Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97)*, 1997, p. 762.

[18] R. O. Stehling, M. A. Nascimento, and A. X. Falcao, "A compact and efficient image retrieval approach based on border/interior pixel classification," in *CIKM '02: Proceedings of the eleventh international conference on Information and knowledge management*, 2002, pp. 102–109.

[19] V. Kovalev and S. Volmer, "Color co-occurence descriptors for querying-by-example," in *MMM '98: Proceedings of the 1998 Conference on MultiMedia Modeling*, 1998, p. 32.

[20] B. Tao and B. W. Dickinson, "Texture recognition and image retrieval using gradient indexing," *Journal of Visual Communication and Image Representation*, vol. 11, no. 3, pp. 327 – 342, 2000.

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

[22] J. van de Weijer and C. Schmid, "Coloring local feature extraction," in *European Conference on Computer Vision*, vol. Part II. Springer, 2006, pp. 334–348.