Unsupervised Similarity Learning through Cartesian Product of Ranking References for Image Retrieval Tasks

Lucas Pascotti Valem and Daniel Carlos Guimarães Pedronette
Department of Statistic, Applied Math. and Computing
Universidade Estadual Paulista (UNESP), Rio Claro, Brazil
lucasvalem@rc.unesp.br, daniel@rc.unesp.br

***Abstract*—Despite the consistent advances in visual features and other Content-Based Image Retrieval techniques, measuring the similarity among images is still a challenging task for effective image retrieval. In this scenario, similarity learning approaches capable of improving the effectiveness of retrieval in an unsupervised way are indispensable. A novel method, called Cartesian Product of Ranking References (CPRR), is proposed with this objective in this paper. The proposed method uses Cartesian product operations based on rank information for exploiting the underlying structure of datasets. Only subsets of ranked lists are required, demanding low computational efforts. An extensive experimental evaluation was conducted considering various aspects, four public datasets and several image features. Besides effectiveness, experiments were also conducted to assess the efficiency of the method, considering parallel and heterogeneous computing on CPU and GPU devices. The proposed method achieved significant effectiveness gains, including competitive state-of-the-art results on popular benchmarks.**

***Keywords*-content-based image retrieval; unsupervised learning; Cartesian product; effectiveness; efficiency;**

I. Introduction

The evolution of technologies for acquisition and sharing digital visual contents has triggered a tremendous growth of image collections. With huge amounts of imagery being accumulated daily from a wide variety of sources [1], Content-Based Image Retrieval (CBIR) systems are considered a central solution for searching and indexing images through their visual content.

Mainly supported by the creation of several visual features and distance measures, the CBIR systems have established itself as an essential tool in many fields. However, despite of the consistent development of image retrieval approaches, effectively measuring the similarity among images remains a challenging task. Thus, more recent approaches have been put efforts on other stages of the retrieval process, not directly related to low-level feature extraction procedures [2].

Several post-processing methods have been proposed to improve the effectiveness of image retrieval tasks in an unsupervised way [3–5]. In general, the main objective of such methods is to replace pairwise distances by more global affinity measures capable of considering the dataset structure [4]. Although effective, approaches based on diffusion processes [3] and graphs [6] require high computational efforts.

In this scenario, rank-based approaches have been attracted a lot of attention lately, considering both effectiveness and efficiency aspects. Various recent methods demonstrated [7–9] that rank analysis can provide a rich and reliable source of information for context-based measures. Rank information have been analyzed through different models, as similarity of ranked lists [7] or sets [8], rank-based recommendations [9], and rank consistency verifications [10]. Regarding efficiency, once the most relevant information is located at top positions of the ranked lists, the computation efforts required can be substantially reduced. Additionally, the rank modeling of similarity information allows an uniform representation, independent of distance or similarity measures.

In this paper, a novel unsupervised similarity learning method is proposed to improve the effectiveness of image retrieval tasks. The main objective of the proposed Cartesian Product of Ranking References (CPRR) is to maximize the similarity information encoded in rankings through Cartesian product operations. While the CPRR algorithm only considers a subset of ranked lists for reducing computational costs, the Cartesian product is used for expanding the similarity relationships. The central idea consists in the use of kNN and reverse kNN queries for computing sets of images, which are used for Cartesian product operations. To the best of our knowledge, this is the first unsupervised similarity learning approach which models the rank information in terms of Cartesian product of neighborhood and reverse neighborhood sets. In addition, the proposed method can be used in rank aggregation tasks and can be efficiently computed through parallel computing.

An extensive experimental evaluation was conducted, considering various aspects. Four public datasets and several different image descriptors are considered. Experimental results confirm the effectiveness of the proposed method, consistently improving the retrieval precision and achieving relative gains up to +32.57%. Besides to the effectiveness, the efficiency of the proposed method was also evaluated. Experiments conducted in parallel and heterogeneous environments (CPUs and GPUs) demonstrated that the algorithm presents very small run times for image collections of different sizes. The CPRR algorithm also compares favorably with recent retrieval methods and state-of-the-art approaches, considering both effectiveness and efficiency aspects. The algorithm achieves a N-S score of 3.93 on the popular UKBench [11] dataset.

The paper is organized as follows: Section II describes the image retrieval model considered; Section III presents the proposed algorithm; Section IV discusses the experimental evaluation; finally, Section V draws the conclusions.

II. Image Retrieval Model

The image retrieval notation used along the paper is formally defined in this section. Let $\mathcal{C}$=$\{img_1, img_2, \ldots, img_n\}$ be an image collection, where n denotes the size of the collection. Let ρ: $\mathcal{C} \times \mathcal{C} \rightarrow \mathbb{R}$ be a similarity function, such that $\rho(img_i, img_j)$ denotes the similarity between two images img_i, $img_j \in \mathcal{C}$. For simplicity and readability purposes, the notation $\rho(i,j)$ is used in the remainder of the paper.

The similarity among all images img_i, $img_j \in \mathcal{C}$ defined by the function $\rho(i,j)$ can be applied for computing an affinity matrix W. The matrix W, in turn, is commonly used as an adjacency matrix by various graph and diffusion-based methods. However, this approach often leads to storage and time complexity of at least $O(n^2)$, so that scalability and efficiency requirements are not met for large image collections.

A rank-based modeling of similarity information represents an effective and efficient solution in this scenario. Different from similarity functions which establish relationships only between pairs of images, the ranked lists encode similarity information among a query image and all other collection images. In addition, although a ranked list can encode information from the entire collection, the most similar images are expected to be located at top positions. Therefore, a constant $L \ll n$ can be used such that only a subset composed of top-L positions of the ranked list are considered, reducing the computational efforts required.

Formally, the ranked list τ_q=(img_1, img_2, $\ldots$, img_L) can be defined as a permutation of the image collection $\mathcal{C}_s \subset \mathcal{C}$, which contains the most similar images to query image img_q, such that and $|\mathcal{C}_L| = L$. A permutation τ_q is a bijection from the set $\mathcal{C}_L$ onto the set $[L] = \{1, 2, \ldots, L\}$. The notation $\tau_q(i)$ can be interpreted as the position (or rank) of image img_i in the ranked list τ_q. If img_i is ranked before img_j in the ranked list of img_q, that is, $\tau_q(i) < \tau_q(j)$, then $\rho(q,i) \geq \rho(q,j)$.

Every image in the collection can be taken as a query image img_q and a respective ranked list can be computed. In this way, the set of ranked lists $\{\tau_1, \tau_2, \ldots, \tau_n\}$ provides a compact and effective rank-based modeling of similarity information. In this work, an unsupervised method is proposed aiming at exploiting the information encoded in the set of ranked lists for computing new and more effective retrieval results.

III. Cartesian Product of Ranking References (CPRR)

The rank analysis have been established as a rich and reliable source of information for context-based measures. The main objective of the proposed Cartesian Product of Ranking References (CPRR) is to maximize the available rank information through the use of Cartesian product operations. The Cartesian product over neighborhood sets establishes new pairwise relationships, which are used to discovering underlying similarity information. The proposed approach can broadly divided in two main steps:

- **Rank Normalization:** the reciprocal rank references are analyzed aiming at improving the symmetry of neighborhoods and, consequently, the effectiveness of the ranked lists;
- **Cartesian Product of Ranking References:** the Cartesian product is computed considering the top-k neighborhood and the reverse neighborhood sets. The obtained results are used to define an iterative similarity measure.

Each step of the proposed approach is discussed in the following subsections.

A. Rank Similarity Score

This section defines a rank similarity score, which is used for both rank normalization and Cartesian product procedures. Since the most relevant information about similarity among images is encoded at top positions of ranked lists, neighborhood sets can be defined at different depths. When considering a given depth d, only the similarity information of top-d ranked images is considered, avoiding noisy information contained in the remainder of ranked lists.

Aiming at considering only the top-d most similar images, a neighborhood set $\mathcal{N}$ is defined. Let d denote the depth of ranked lists considered, and therefore the size of the neighborhood set. Let $\mathcal{N}(i,d)$ be the neighborhood set, which is formally defined as follows:

$$\mathcal{N}(q,d) = \{\mathcal{R} \subseteq \mathcal{C}, |\mathcal{R}| = d \wedge \forall x \in \mathcal{R}, y \in \mathcal{C} - \mathcal{R} : \rho(q,x) \geq \rho(q,y)\}. \tag{1}$$

Taking into account the neighborhood set, a similarity score is defined based on rank information. The rank similarity score $r_d(q,i)$ represents the similarity between images img_q and img_i based on the ranked list of image img_q at a depth d. The score ranges linearly according to the position of img_i in the ranked list τ_q and can be defined as:

$$r_d(q,i) = \begin{cases} d - \tau_q(i) + 1 & if\ img_i \in \mathcal{N}(q,d) \\ 0 & otherwise. \end{cases} \tag{2}$$

In this work, two different depths are considered: L, which defines a broader neighborhood used by the rank normalization step; and k, which defines a local neighborhood used by Cartesian product operations. A sparse matrix structure [9] can be used for storage of scores, once both k and L values are much smaller than n.

B. Reciprocal Rank Normalization

Different from most of distance or similarity measures, the k-neighborhood relationships and rank measures are not symmetric. However, the benefits of improving the symmetry of the k-neighborhood relationship are remarkable in image retrieval applications [12]. In this way, various approaches have been proposed for exploiting the reciprocal neighborhood [13].

In this work, a simple rank normalization based on the reciprocal neighborhood is employed as a pre-processing step for the Cartesian product operations. A normalized similarity function $\rho_r(i,j)$ is defined as the sum of reciprocal rank similarity score at a depth L:

$$\rho_r(i,j) = r_L(i,j) + r_L(j,i). \tag{3}$$

In the following, all the ranked lists are updated according to the similarity function, using a stable sorting algorithm. The update gives rise to a new set of ranked lists, which is used as input for the Cartesian product operations. Notice that a low computational cost algorithm can be derived for computing the rank normalization procedure, once only the top-L positions of ranked lists are considered.

C. Cartesian Product of Neighborhood Sets

The ranked lists and the neighborhood sets represents a relevant source of information for context-based similarity functions. While a pairwise measure defines a similarity relation between only two images, a ranked list establishes a broader relationship among a query image and its most similar images. Additionally, the neighborhood and rank analysis can be exploited for discovering underlying similarity information among neighbors of a same query image.

In this way, Cartesian product operations of neighborhood sets and rank information are employed for computing a new and more effective similarity measure. The objective is to consider the pairwise relations computed by the Cartesian product weighted by rank information.

The Cartesian product can be defined as the set of all possible pairs of elements whose components are members of two sets. Let $\mathcal{N}(i,k)$ and $\mathcal{N}(j,k)$ be k-neighborhood sets of images img_i and img_j, respectively. The Cartesian product of $\mathcal{N}(i,k)$ and $\mathcal{N}(j,k)$ is defined as:

$$\mathcal{N}(i,k) \times \mathcal{N}(j,k) = \{(n_i, n_j) \mid n_i \in \mathcal{N}(i,k),\ n_j \in \mathcal{N}(j,k)\}. \tag{4}$$

Given a query image img_q and its respective neighborhood set $\mathcal{N}(q,k)$, we denote $\mathcal{N}(q,k) \times \mathcal{N}(q,k) = \mathcal{N}(q,k)^2$. The pairs of images contained in $\mathcal{N}(q,k)^2$ define all possible similarity relationships among the neighbors of img_q. This information is exploited for computing a new similarity score $wc(i,j)$, between img_i and img_j, with $img_i, img_j \in \mathcal{N}(q,k)$.

The similarity score $wc(i,j)$ is computed considering every query image $img_q \in \mathcal{C}$ which have img_i and img_j as neighbors. In addition, the score is weighted according to their rank score, by the term $r_k(q,i) \times r_k(q,j)$. Formally, the score $wc(i,j)$ can be defined as:

$$wc(i,j) = \sum_{q \in \mathcal{C}} \sum_{i \in \mathcal{N}(q,k)^2} \sum_{j \in \mathcal{N}(q,k)^2} r_k(q,i) \times r_k(q,j). \tag{5}$$

Algorithmically, the similarity score can be computed with complexity of only $O(n)$, once k is a constant. Algorithm 1 outlines our proposed approach. The main idea consists in performing only top-k rank analysis for computing the Cartesian product for each neighborhood set.

Algorithm 1 Cartesian Product of Neighborhood Sets.

```
Require: Set of Ranked Lists $\mathcal{R}$
Ensure: Similarity score $wc(\cdot,\cdot)$
for all $img_q \in \mathcal{C}$ do
    for all $img_i \in \mathcal{N}(q,k)$ do
        for all $img_j \in \mathcal{N}(q,k)$ do
            $wc(i,j) \leftarrow wc(i,j) + r_k(q,i) \times r_k(q,j)$
            $wc(j,i) \leftarrow wc(j,i) + r_k(q,i) \times r_k(q,j)$
        end for
    end for
end for
```

D. Cartesian Product of Reverse Neighborhood Sets

The Cartesian product of ranked lists defines similarity relationships among neighbors of the same query image. On the other hand, information from different queries with a common neighbor is ignored. In other words, the set of images which have an image among its neighbors also encodes a relevant similarity information, which can be exploited for improving the effectiveness of retrieval.

With this objective, the Cartesian product of reverse neighborhood sets is considered. Let $\mathcal{N}_r(q)$ be a reverse neighborhood set computed for an image img_q, which is compose by all images whose neighborhood sets contains img_q. Formally, the set $\mathcal{N}_r(q)$ can be defined as follows:

$$\mathcal{N}_r(x) = \{\mathcal{R} \subseteq \mathcal{C}, \forall q \in \mathcal{R} : img_x \in \mathcal{N}(q,k)\}. \tag{6}$$

The Cartesian product of the reverse neighborhood set $\mathcal{N}_r(i)^2$ is used for analysing underlying similarity information. In this way, a similarity score $wr(i,j)$ is defined for increasing the similarity between any given images img_i and img_j contained in reverse neighborhood sets. Formally, the score is defined as follows:

$$wr(i,j) = \sum_{x \in \mathcal{C}} \sum_{i \in \mathcal{N}_r(x)^2} \sum_{j \in \mathcal{N}_r(x)^2} r_k(i,x) \times r_k(j,x). \tag{7}$$

An algorithmic solution for computing the similarity score based on reverse neighborhood is presented in Algorithm 2. The reverse neighborhood sets are computed on lines 1-5, while the Cartesian product is computed on lines 6-13.

E. Iterative Similarity Measure

The similarity scores based on the Cartesian product of neighborhood and reverse neighborhood sets are used for computing a new and iterative similarity measure. Let the superscript $^{(t)}$ denote the current iteration, the similarity measure $\rho^{(t+1)}$ can be defined as:

$$\rho^{(t+1)} = wc(i,j) + wr(i,j). \tag{8}$$

The similarity measure $\rho^{(t+1)}$ is used as input for a sorting step, which gives rise to a new set of ranked lists. Once the input of algorithm is also a set of ranked lists, it can be iteratively executed until a certain number of T iterations.

Algorithm 2 Cartesian Product of Reverse Neighborhood Sets.

```
Require: Set of Ranked Lists $\mathcal{R}$
Ensure: Similarity score $wr(\cdot,\cdot)$
for all $img_q \in \mathcal{C}$ do
    for all $img_x \in \mathcal{N}(q,k)$ do
        $\mathcal{N}_r(x) \leftarrow \mathcal{N}_r(x) \cup img_q$
    end for
end for
for all $img_x \in \mathcal{C}$ do
    for all $img_i \in \mathcal{N}_r(x)$ do
        for all $img_j \in \mathcal{N}_r(x)$ do
            $wr(i,j) \leftarrow wc(i,j) + r_k(i,x) \times r_k(j,x)$
            $wr(j,i) \leftarrow wc(j,i) + r_k(i,x) \times r_k(j,x)$
        end for
    end for
end for
```

Only the Cartesian product operations are considered for the iterative measure, e.g., the rank normalization is executed only before the first iteration. The method also requires a very small number of iterations for reaching high effectiveness results (as discussed in Section IV).

F. Rank Aggregation

Different features often provide distinct visual information about images. In this scenario, different rankings computed for each feature also encode distinct and complementary information. In fact, most of recent retrieval approaches commonly consider various features [14]. Our goal is to use the proposed CPRR algorithm for rank aggregation tasks, aiming at combining rank information computed for different features.

Since the most significant effectiveness gains are obtained at the first iteration, the CPRR algorithm is computed independently for each descriptor and the similarity measure is combined before the next iterations. Let $\rho_a^{(1)}$ be the similarity measure computed at the first iteration for a given feature f_a. Let a be defined in the interval $[1, m]$, where m denotes number of features considered. The combined similarity measure is computed as follows:

$$\rho^{(1)}(i,j) = \sum_{a=1}^{m} \rho_a^{(1)}(i,j). \tag{9}$$

Based on similarity score, a new set of ranked lists is generated. Subsequently, the next iterations are computed in the same way as the similarity learning algorithm.

G. Parallel Design

The CPRR algorithm can be widely parallelized, specially regarding its Cartesian product operations. This section discusses the parallel design of the CPRR algorithm, using the OpenCL standard. The OpenCL is a low-level API for task-parallel and data-parallel heterogeneous computing. A *kernel* is the name given for pieces of code that can be executed in parallel. Each kernel is executed in parallel by a given number of *work-items*.

The parallel design of the CPRR algorithm is illustrated in Figure 1. Each main step of the algorithm defines a different kernel, which runs in an OpenCL device (CPU or GPU). Each kernel, in turn, is parallelized in n work-items. Two transfer models were used: *Writer Buffer*, which requires the transfer of the data to the device memory; and *Map Buffer*, which requires only the transfer of data pointers.

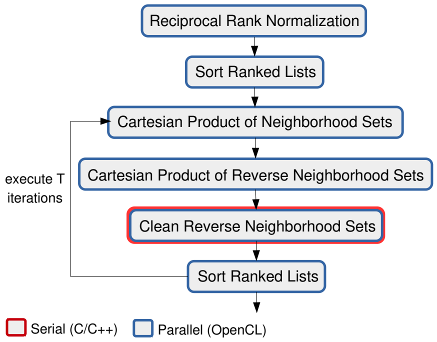
Reciprocal Rank Normalization
Sort Ranked Lists
Cartesian Product of Neighborhood Sets
Cartesian Product of Reverse Neighborhood Sets
Clean Reverse Neighborhood Sets
Sort Ranked Lists
execute T iterations
Serial (C/C++)
Parallel (OpenCL)

Fig. 1: Design of the Parallel CPRR Algorithm.

Since the Cartesian product operations are processed in parallel and the similarity measure is stored using a global memory, concurrent access can cause loss of updates. However, the overhead associated to atomic operations in OpenCL is high. Therefore, direct updates of similarity score are allowed due to the very low impact on the effectiveness of the algorithm (as discussed in the experimental section).

Notice that almost all the steps are executed in parallel. However, the "Clean Reverse Neighborhood Sets" can either be serial or parallel. When using Write Buffer, this step is executed in parallel to avoid the transfer to device memory, which causes efficiency loss. For the Map Buffer, this operation can be executed serially without any transfer before and after the operation.

In the end of each iteration, a sorting procedure is executed to update the top positions of ranked lists according to the new similarity measure. The insertion sort algorithm is used, once it tends to be linear when input is almost sorted.

IV. Experimental Evaluation

Several aspects are considered to assess the effectiveness and efficiency of the proposed method. Section IV-A describes the experimental setup. Section IV-B discusses the algorithm settings. Section IV-C presents the results of the effectiveness evaluation, while Section IV-D presents the efficiency evaluation. Section IV-E presents a comparison of the proposed method with other state-of-the-art unsupervised learning methods and recent retrieval approaches.

A. Experimental Setup

Four distinct and public datasets are considered in the experiments, with size ranging from 280 to 10,200 images. In order to exploit the different dataset characteristics, several global descriptors are used, considering shape, color, and texture properties. Convolution Neural Network features are extracted using the Caffe framework [37] and different layers

TABLE I: Datasets and images descriptors used in the experimental evaluation.

Dataset	Size	Type	General Description	Descriptors	Effectiv. Measure
Soccer [15]	280	Color Scenes	Dataset composed of images from 7 soccer teams, containing 40 images per class.	Border/Interior Pixel Classification (BIC) [16], Auto Color Correlograms (ACC) [17], and Global Color Histogram (GCH) [18]	MAP (%)
MPEG-7 [19]	1,400	Shape	A well-known dataset composed of 1400 shapes divided in 70 classes. Commonly used for evaluation of unsupervised learning approaches.	Segment Saliences (SS) [20], Beam Angle Statistics (BAS) [21], Inner Distance Shape Context (IDSC) [22], Contour Features Descriptor (CFD) [23], Aspect Shape Context (ASC) [24], and Articulation-Invariant Representation (AIR) [25]	MAP (%), Recall@40
Brodatz [26]	1,776	Texture	A popular dataset for texture descriptors evaluation composed of 111 different textures divided into 16 blocks.	Local Binary Patterns (LBP) [27], Color Co-Occurrence Matrix (CCOM) [28], and Local Activity Spectrum (LAS) [29]	MAP (%)
UKBench [11]	10,200	Objects/ Scenes	Composed of 2,550 objects or scenes. Each object/scene is captured 4 times from different viewpoints, distances, and illumination conditions.	ACC [17], ACC Spatial Pyramid (ACC-SPy) [17, 30], BIC [16], Color and Edge Directivity Descriptor (CEED) [31], Fuzzy Color and Texture Histogram (FCTH) [32], FCTH Spatial Pyramid (FCTH-SPy) [30, 32], Joint Composite Descriptor (JCD) [33], Scale-Invariant Feature Transform (SIFT) [34], Scalable Color Descriptor (SCD) [35], Vocabulary Tree (VOC) [36], and Convolutional Neural Network by Caffe [37] framework (CNN-Caffe)	N-S Score

(FC6, FC7, FC8). Regarding local descriptors, SIFT [34] features are used considering a variant of vocabulary tree based retrieval (VOC) [36]. The datasets and descriptors are briefly described in Table I.

The effectiveness evaluation considers all images of each dataset as query images. As effectiveness measure, the Mean Average Precision (MAP) is used for most of the datasets, except for the UKBench dataset [11] which uses the N-S Score. For the MPEG-7 [19] dataset, the Recall@40 is also considered in addition to MAP. Most of experiments also report the effectiveness gains: let S_b and S_a be the effectiveness scores before and after the algorithm execution, the gain is defined as: $(S_a - S_b)/S_b$.

For the efficiency evaluation experiments, the average run time of 10 executions and 95% confidence intervals are considered. The hardware environment is composed of a CPU Intel Xeon CPU E3-1240 and a GPU AMD Radeon HD 7900. The software environment is given by the operating system Linux 3.11.0-15 - Ubuntu 12.04 and OpenCL 1.2 AMD-APP. The code was compiled using g++ 4.6.3 using the flag "-O3".

B. Parameter Settings

Two parameters are considered in the proposed algorithm: *(i)* k: the number of nearest neighbors; and *(ii)* T: the number of iterations. Additionally, the constant L defines a trade-off between effectiveness and efficiency. A set of experiments were conducted for evaluating the impact of different parameter settings on the retrieval scores.

The first experiment aims at analyzing the impact of different combination of parameters k and T. The MAP scores are computed ranging the parameter k in the interval $[0, 30]$ and the parameter T from 1 to 5. This experiment considers the MPEG-7 [19] dataset and the CFD [23] shape descriptor. Figure 2 shows the variation of the MAP according to k and T. The joined growth of k and MAP scores can be observed until a stabilization, with values of k near 20. For the parameter T, the most significant effectiveness gains are obtained for the first iteration. Considering these results, the parameter values of $k = 20$ and $T = 2$ were used for most of the remaining experiments. Only the UKBench [11] dataset used $k = 4$. This is due to the very distinct size of classes: the dataset has a very small number of images per class.

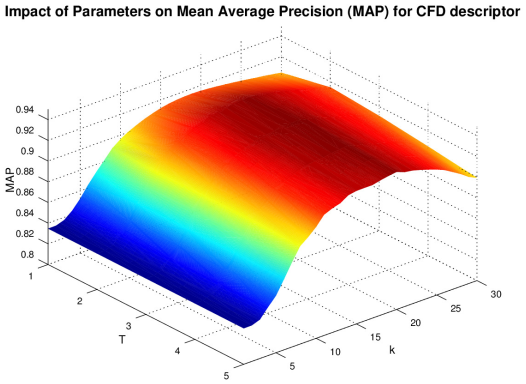

Fig. 2: Impact of parameters k and T on effectiveness.

Since the CPRR algorithm does not require the use of the entire ranked list, the second experiment analyzes the impact of the size of ranked lists on the effectiveness results. The higher the L value, the greater the effectiveness but also the greater the run time required. The experiment conducted on the MPEG-7 [19] dataset analyzed the MAP scores according to different values of L ranging in the interval $[50, 1400]$. The results for four different descriptors are shown in Figure 3. As it can be observed, the most significant gains are obtained for small values of L. In addition, for higher values of L, the MAP scores reach an asymptote. In this way, the value of $L = 400$ was used for most of the experiments.[1]

C. Effectiveness Evaluation

Various experiments were conducted aiming at evaluating the effectiveness of the proposed CPRR algorithm. Diverse public datasets, several descriptors and various baselines were considered.

Firstly, the proposed algorithm is evaluated in generic image retrieval tasks, considering three datasets and different global features (shape, color, and texture). The MAP results are shown in Table II, considering both serial and parallel implementations of CPRR algorithm. As it can be observed, the effectiveness results for serial and parallel implementations are very similar. The relative effectiveness gains, computed based on serial execution, achieve very high values up to +32.57%. The results of recent unsupervised learning approaches [9, 38] are reported as baselines. We can observe that the proposed CPRR algorithm yields the best scores for most of descriptors.

[1] The value of L used for the Soccer [15] dataset is limited by the dataset size ($L = 280$). For the UKBench [11] dataset we used $L = 200$.

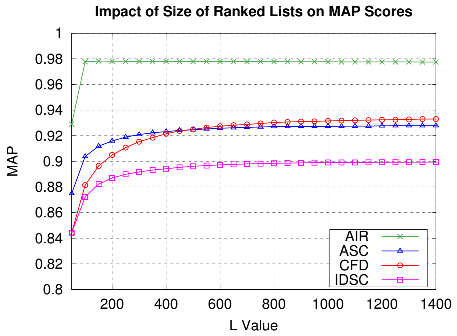

Fig. 3: Impact of L on effectiveness.

TABLE II: Effectiveness evaluation of the proposed CPRR algorithm considering various datasets, descriptors, and baselines (MAP as effectiveness measure).

Descriptor	Dataset	Original MAP	Pairwise Rec. [38]	RL-Rec. [9]	CPRR Serial	CPRR Parallel	Gain
SS [20]	MPEG-7	37.67%	39.90%	48.68%	49.94%	49.947 % ± 0.009	+32.57%
BAS [21]	MPEG-7	71.52%	77.65%	79.58%	80.60%	80.611 % ± 0.006	+12.70%
IDSC [22]	MPEG-7	81.70%	86.83%	88.80%	89.42%	89.432 % ± 0.004	+9.45%
CFD [23]	MPEG-7	80.71%	91.38%	91.39%	92.15%	92.157 % ± 0.005	+14.17%
ASC [24]	MPEG-7	85.28%	91.80%	91.34%	92.32%	92.323 % ± 0.005	+8.26%
AIR [25]	MPEG-7	89.39%	95.50%	96.12%	97.80%	97.796 % ± 0.007	+9.41%
GCH [18]	Soccer	32.24%	32.35%	34.38%	35.47%	35.307 % ± 0.054	+10.02%
ACC [17]	Soccer	37.23%	40.31%	41.23%	47.14%	46.965 % ± 0.072	+26.62%
BIC [16]	Soccer	39.26%	42.64%	45.15%	47.29%	47.172 % ± 0.095	+20.45%
LBP [27]	Brodatz	48.40%	51.92%	51.26%	49.07%	49.073 % ± 0.006	+1.38%
CCOM [28]	Brodatz	57.57%	66.46%	64.34%	64.81%	64.816 % ± 0.007	+12.58%
LAS [29]	Brodatz	75.15%	80.73%	79.71%	79.34%	79.346 % ± 0.004	+5.58%

TABLE III: Effectiveness evaluation of the CPRR algorithm on the UKBench [11] dataset, considering the N-S score.

Descriptor	Type	Original Score	RL-Rec. [9]	CPRR	Gain
SIFT [34]	Local	2.54	2.88	2.99	+17.72%
CEED [31]	Color/Text.	2.61	2.72	2.83	+8.43%
FCTH [32]	Color/Text.	2.73	2.80	2.90	+6.23%
JCD [33]	Color/Text.	2.79	2.88	3.00	+7.53%
FCTH-Spy [30, 32]	Color/Text.	2.91	3.05	3.21	+10.31%
BIC [16]	Color	3.04	3.15	3.28	+7.89%
Caffe-FC6 [37]	CNN	3.05	3.30	3.40	+11.48%
Caffe-FC8 [37]	CNN	3.18	3.30	3.47	+9.12%
ACC-Spy [17, 30]	Color	3.25	3.38	3.52	+8.31%
Caffe-FC7 [37]	CNN	3.31	3.46	3.61	+9.06%
ACC [17]	Color	3.36	3.53	3.62	+7.74%
SCOLOR [34]	Color	3.15	3.24	3.37	+6.98%
VOC [36]	BoW	3.54	3.65	3.72	+5.08%

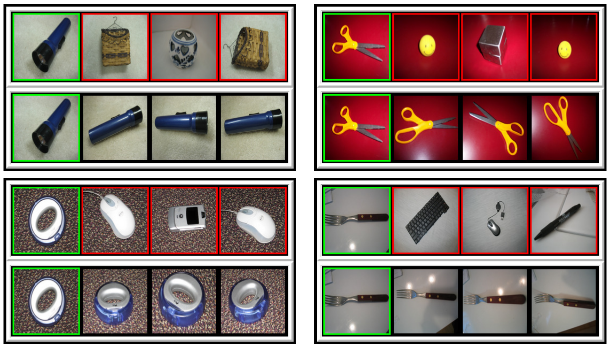

Fig. 4: Impact of the CPRR on the UKBench [11] dataset.

TABLE IV: Effectiveness evaluation of rank aggregation tasks.

Dataset	Descriptors	Score	Metric
MPEG-7	CFD+AIR	99.95%	MAP
MPEG-7	CFD+ASC	98.83%	MAP
Brodatz	CCOM+LAS	83.26%	MAP
Soccer	ACC+BIC	48.25%	MAP
UKBench	VOC+CNN-FC7	3.88	N-S score
UKBench	ACC+CNN-FC7	3.88	N-S score
UKBench	ACC+CNN-FC7+VOC	3.93	N-S score

An experiment considering natural image retrieval tasks and a very distinct set of descriptors was conducted on the UKBench [11] dataset. Table III presents the effectiveness results given by the N-S score. The N-S score corresponds to the number of relevant images among the first four images returned, defined in the interval [1,4] (the highest achievable score is 4). The small number of images per class (only 4) makes this dataset a very challenging one for unsupervised learning algorithms. Despite this fact, the CPRR achieved high gains ranging from +5.08% to +17.72% and superior to recent baseline [9].

Figure 4 illustrates four visual examples of the impact of the CPRR algorithm on retrieval results obtained for the UKBench [11] dataset and the ACC [17] descriptor. The query images are presented in green borders and wrong results in red borders. The first line represents the original retrieval results and the second line, the results after the algorithm execution.

The proposed algorithm was also evaluated in rank aggregation tasks, considering the best descriptors for each dataset. Table IV presents the effectiveness scores for various datasets. A slightly different parameter settings[2] were used due to the possibility of relevant images do not appear in the top-L positions of all combined descriptors. We can observe that all aggregated results are superior to isolated descriptors, reaching very high scores for all datasets.

An experiment analyzing both effectiveness and efficiency aspects was conducted on the MPEG-7 [19] dataset. Figure 5 presents the results of CPRR and recent baselines (Pairwise Recommendation [38], RL-Sim [7, 39] and RL-Recommendation [9]). The MAP score and the run time determines the position of the algorithms in the graph. Therefore, an ideal algorithm, with high effectiveness and low run time, is positioned at the top-left corner of the graph. Notice that the best positions are occupied by the CPRR Algorithm (serial and parallel).

D. Efficiency Evaluation

Experiments were conducted for evaluating the efficiency of the proposed method, considering several aspects, as: different datasets, serial and parallel[3] implementations, different devices (CPU, GPU), and memory transfer models.

Table V presents the average run time and confidence intervals for the CPRR algorithm considering different criteria. For comparison purposes, the run time of two recent baselines [9, 38] are reported. The best performance for each dataset is

[2] We used L = 600 for the MPEG-7 [19] and Brodatz [26]. For the UKBench [11] dataset, we used L = 100 and k = 6.

[3] The OpenCL build and environment time are not considered in the reported results, since the build can be executed once off-line and the environment time is constant independently of dataset sizes.

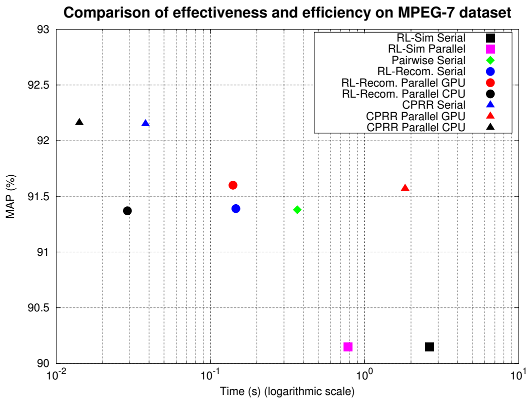

Fig. 5: Effectiveness and efficiency analysis.

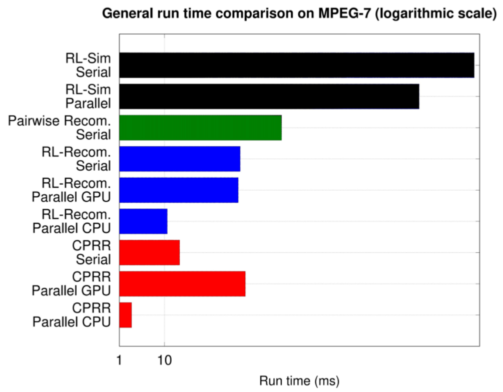

Fig. 6: Run time comparison on the MPEG-7 [19] dataset.

TABLE V: Efficiency evaluation: runtime (in seconds) of the CPRR for different devices and datasets.

Algorithm	Exec.	Device	Soccer [15]	MPEG-7 [19]	Brodatz [26]	UKBench [11]
Pairwise Rec. [38]	Serial	CPU	0.1149 ± 0.00018	0.3663 ± 0.00094	0.6672 ± 0.00140	14.802 ± 0.11059
RL-Rec. [9]	Serial	CPU	0.0607 ± 0.00000	0.1462 ± 0.00021	0.1108 ± 0.00102	0.1868 ± 0.00018
CPRR	Serial	CPU	0.0058 ± 0.00021	0.0381 ± 0.00041	0.0501 ± 0.00077	0.1767 ± 0.00102
CPRR	Parallel	GPU[1]	0.0711 ± 0.00463	0.1640 ± 0.00781	0.1560 ± 0.00570	0.2038 ± 0.00874
CPRR	Parallel	CPU[1]	0.0032 ± 0.00015	0.0164 ± 0.00031	0.0214 ± 0.00054	0.1834 ± 0.00052
CPRR	Parallel	CPU[2]	**0.0027 ± 0.00000**	**0.0131 ± 0.00018**	**0.0143 ± 0.00075**	**0.1082 ± 0.00051**

Memory Transfer Model: [1]Write Buffer; [2]Map Buffer.

highlighted in bold. As we can observe, the CPRR algorithm requires very low run times for all datasets, smaller than baselines even for the serial implementation.

A more general comparison of the CPRR (both serial and parallel) is presented in Figure 6. The run time of the RL-Sim [7, 39] (serial and parallel), RL-Recommendation [9] (serial and parallel) and Pairwise Recommendation [38] (serial), are reported as baselines. The MPEG-7 [19] dataset is considered for the experiment. Notice that, even using a logarithmic scale, the run time of the proposed CPRR (in red) algorithm is significant smaller than other considered approaches.

E. Comparison with Other Approaches

Finally, the CPRR algorithm was also evaluated in comparison with other state-of-the-art unsupervised learning methods and recently proposed retrieval approaches. Experiments were conducted on two image datasets: UKBench [11] and MPEG-7 [19], which are popular datasets commonly used as benchmark for image retrieval and post-processing methods.

Table VI presents the effectiveness results of CPRR algorithm in comparison with recent retrieval approaches on the UKBench [11] dataset. We can observe that the CPRR algorithm achieves the best results, reaching a N-S score of **3.93** for the aggregation of VOC+ACC+CNN-FC7 features.

A comparison of the proposed method with other state-of-the-art unsupervised methods on the MPEG-7 [19] dataset is shown in Table VII. The effectiveness results obtained by the CPRR algorithm are also comparable or superior to various other approaches.

TABLE VI: Effectiveness comparison among recent retrieval methods on the UKBench [11] dataset.

N-S scores for recent retrieval methods					
Zheng *et al.* [40]	Qin *et al.* [13]	Wang *et al.* [41]	Zhang *et al.* [14]	Zheng *et al.* [42]	Xie *et al.* [43]
3.57	3.67	3.68	3.83	3.84	3.89

N-S scores for the CPRR method		
ACC+CNN-FC7	VOC+CNN-FC7	VOC+ACC+CNN-FC7
3.88	3.88	**3.93**

TABLE VII: Comparison of post-processing methods on the MPEG-7 [19] dataset - Bull's Eye Score (Recall@40).

Shape Descriptors		
CFD [23]	-	84.43%
IDSC [22]	-	85.40%
SC [44]	-	86.80%
ASC [24]	-	88.39%
AIR [25]	-	93.67%
Post-Processing Methods		
Algorithm	**Descriptor(s)**	**Score**
Graph Transduction [45]	IDSC	91.00%
Shortest Path Propagation [6]	IDSC	93.35%
RL-Sim [7]	CFD	94.13%
CPRR	**CFD**	**94.77%**
Locally C. Diffusion Process [3]	ASC	95.96%
RL-Recommendation [9]	ASC	94.40%
CPRR	**ASC**	**95.07%**
Tensor Product Graph [4]	ASC	96.47%
Self-Smoothing Operator [5]	SC+IDSC	97.64%
Self-Smoothing Operator [5]	SC+IDSC+DDGM	99.20%
CPRR	**CFD+ASC**	**99.51%**
Pairwise Recommendation [38]	CFD+IDSC	99.52%
RL-Recommendation [9]	AIR	99.78%
CPRR	**AIR**	**99.93%**
Tensor Product Graph [4]	AIR	99.99%
Neighbor Set Similarity [8]	AIR	100%
CPRR	**CFD+AIR**	**100%**

V. Conclusions

In this paper, we presented a novel unsupervised similarity learning algorithm for image retrieval tasks. The proposed approach employs Cartesian product operations for analyzing rank information and exploiting the underlying structure of the datasets. Extensive experiments were conducted considering public datasets and several descriptors. The experimental results and comparisons with other recent state-of-the-art approaches demonstrate the effectiveness and efficiency of the proposed method. As future work, we intend to investigate the use of the proposed method in semi-supervised learning tasks, considering interactive image retrieval scenarios. We also intend to investigate its use in scenarios where the query image is not part of the dataset.

Acknowledgment

The authors are grateful to São Paulo Research Foundation - FAPESP (grants 2013/08645-0 and 2014/04220-8).

References

[1] B. Thomee and M. Lew, "Interactive search in image retrieval: a survey," *International Journal of Multimedia Information Retrieval*, vol. 1, no. 2, pp. 71–86, 2012.

[2] Y. Liu, D. Zhang, G. Lu, and W.-Y. Ma, "A survey of content-based image retrieval with high-level semantics," *Pattern Recognition*, vol. 40, no. 1, pp. 262 – 282, 2007.

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

[4] X. Yang, L. Prasad, and L. Latecki, "Affinity learning with diffusion on tensor product graph," *IEEE TPAMI*, vol. 35, no. 1, pp. 28–38, 2013.

[5] J. Jiang, B. Wang, and Z. Tu, "Unsupervised metric learning by self-smoothing operator," in *ICCV*, 2011, pp. 794–801.

[6] J. Wang, Y. Li, X. Bai, Y. Zhang, C. Wang, and N. Tang, "Learning context-sensitive similarity by shortest path propagation," *Pattern Recognition*, vol. 44, no. 10-11, pp. 2367–2374, 2011.

[7] D. C. G. Pedronette and R. da S. Torres, "Image re-ranking and rank aggregation based on similarity of ranked lists," *Pattern Recognition*, vol. 46, no. 8, pp. 2350–2360, 2013.

[8] X. Bai, S. Bai, and X. Wang, "Beyond diffusion process: Neighbor set similarity for fast re-ranking," *Information Sciences*, vol. 325, pp. 342 – 354, 2015.

[9] L. P. Valem, D. C. G. Pedronette, R. d. S. Torres, E. Borin, and J. Almeida, "Effective, efficient, and scalable unsupervised distance learning in image retrieval tasks," *ICMR*, 2015.

[10] Y. Chen, X. Li, A. Dick, and R. Hill, "Ranking consistency for image matching and object retrieval," *Pattern Recognition*, vol. 47, no. 3, pp. 1349 – 1360, 2014.

[11] D. Nistér and H. Stewénius, "Scalable recognition with a vocabulary tree," in *CVPR*, vol. 2, 2006, pp. 2161–2168.

[12] H. Jegou, C. Schmid, H. Harzallah, and J. Verbeek, "Accurate image search using the contextual dissimilarity measure," *PAMI*, vol. 32, no. 1, pp. 2–11, 2010.

[13] D. Qin, S. Gammeter, L. Bossard, T. Quack, and L. van Gool, "Hello neighbor: Accurate object retrieval with k-reciprocal nearest neighbors," in *CVPR*, june 2011, pp. 777 –784.

[14] S. Zhang, M. Yang, T. Cour, K. Yu, and D. Metaxas, "Query specific rank fusion for image retrieval," *IEEE TPAMI*, vol. 37, no. 4, pp. 803–815, April 2015.

[15] J. van de Weijer and C. Schmid, "Coloring local feature extraction," in *ECCV*.

[16] R. O. Stehling, M. A. Nascimento, and A. X. Falcão, "A compact and efficient image retrieval approach based on border/interior pixel classification," in *CIKM*, 2002, pp. 102–109.

[17] J. Huang, S. R. Kumar, M. Mitra, W.-J. Zhu, and R. Zabih, "Image indexing using color correlograms," in *CVPR*, 1997, pp. 762–768.

[18] M. J. Swain and D. H. Ballard, "Color indexing," *International Journal on Computer Vision*, vol. 7, no. 1, pp. 11–32, 1991.

[19] L. J. Latecki, R. Lakmper, and U. Eckhardt, "Shape descriptors for non-rigid shapes with a single closed contour," in *CVPR*, 2000, pp. 424–429.

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

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

[22] H. Ling and D. W. Jacobs, "Shape classification using the inner-distance," *IEEE TPAMI*, vol. 29, no. 2, pp. 286–299, 2007.

[23] D. C. G. Pedronette and R. da S. Torres, "Shape retrieval using contour features and distance optmization," in *VISAPP*, vol. 1, 2010, pp. 197 – 202.

[24] H. Ling, X. Yang, and L. J. Latecki, "Balancing deformability and discriminability for shape matching," in *ECCV*, vol. 3, 2010, pp. 411–424.

[25] R. Gopalan, P. Turaga, and R. Chellappa, "Articulation-invariant representation of non-planar shapes," in *ECCV*, vol. 3, 2010, pp. 286–299.

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

[27] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *PAMI*, vol. 24, no. 7, pp. 971–987, 2002.

[28] V. Kovalev and S. Volmer, "Color co-occurence descriptors for querying-by-example," in *ICMM*, 1998, p. 32.

[29] B. Tao and B. W. Dickinson, "Texture recognition and image retrieval using gradient indexing," *JVCIR*, vol. 11, no. 3, pp. 327–342, 2000.

[30] M. Lux, "Content based image retrieval with LIRe," in *ACM MM '11*, 2011.

[31] S. A. Chatzichristofis and Y. S. Boutalis, "Cedd: color and edge directivity descriptor: a compact descriptor for image indexing and retrieval," in *ICVS*, 2008, pp. 312–322.

[32] ——, "Fcth: Fuzzy color and texture histogram - a low level feature for accurate image retrieval," in *WIAMIS*, 2008, pp. 191–196.

[33] K. Zagoris, S. Chatzichristofis, N. Papamarkos, and Y. Boutalis, "Automatic image annotation and retrieval using the joint composite descriptor," in *PCI*, 2010, pp. 143–147.

[34] D. Lowe, "Object recognition from local scale-invariant features," in *ICCV*, 1999, pp. 1150–1157.

[35] B. Manjunath, J.-R. Ohm, V. Vasudevan, and A. Yamada, "Color and texture descriptors," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, no. 6, pp. 703–715, 2001.

[36] X. Wang, M. Yang, T. Cour, S. Zhu, K. Yu, and T. Han, "Contextual weighting for vocabulary tree based image retrieval," in *ICCV'2011*, Nov 2011, pp. 209–216.

[37] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," *arXiv preprint arXiv:1408.5093*, 2014.

[38] D. C. G. Pedronette and R. da S. Torres, "Exploiting pairwise recommendation and clustering strategies for image re-ranking," *Information Sciences*, vol. 207, pp. 19–34, 2012.

[39] D. C. G. Pedronette, R. da S. Torres, E. Borin, and M. Breternitz, "Rl-sim algorithm acceleration on GPUs," in *SBAC*, 2013.

[40] L. Zheng, S. Wang, and Q. Tian, "Lp-norm idf for scalable image retrieval," *IEEE TIP*, vol. 23, no. 8, pp. 3604–3617, Aug 2014.

[41] B. Wang, J. Jiang, W. Wang, Z.-H. Zhou, and Z. Tu, "Unsupervised metric fusion by cross diffusion," in *CVPR*, 2012, pp. 3013 –3020.

[42] L. Zheng, S. Wang, L. Tian, F. He, Z. Liu, and Q. Tian, "Query-adaptive late fusion for image search and person re-identification," in *CVPR*, 2015.

[43] L. Xie, R. Hong, B. Zhang, and Q. Tian, "Image classification and retrieval are one," in *ACM ICMR'2015*, 2015, pp. 3–10.

[44] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *IEE TPAMI*, vol. 24, no. 4, pp. 509–522, 2002.

[45] X. Yang, X. Bai, L. J. Latecki, and Z. Tu, "Improving shape retrieval by learning graph transduction," in *ECCV*, vol. 4, 2008, pp. 788–801.