Accepted version of paper published in IEEE Transactions on Image Processing, 2019.
DOI: http://doi.org/10.1109/TIP.2019.2920526

Multimedia Retrieval through Unsupervised Hypergraph-based Manifold Ranking

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

[1]Department of Statistics, Applied Math. and Computing, State University of São Paulo (UNESP), Rio Claro, SP, Brazil
[2]Institute of Science and Technology, Federal University of São Paulo (UNIFESP), São José dos Campos, SP, Brazil
[3]RECOD Lab, Institute of Computing, University of Campinas (UNICAMP), Campinas, SP, Brazil

***Abstract*—Accurately ranking images and multimedia objects is of paramount relevance in many retrieval and learning tasks. Manifold learning methods have been investigated for ranking mainly due to its capacity of taking into account the intrinsic global manifold structure. In this paper, a novel manifold ranking algorithm is proposed based on hypergraphs for unsupervised multimedia retrieval tasks. Different from traditional graph-based approaches, which represents only pairwise relationships, hypergraphs are capable of modeling similarity relationships among set of objects. The proposed approach uses the hyperedges for constructing a contextual representation of data samples, and exploits the encoded information for deriving a more effective similarity function. An extensive experimental evaluation was conducted on nine public datasets, including diverse retrieval scenarios and multimedia content. Experimental results demonstrate that high effectiveness gains can be obtained in comparison with state-of-the-art methods.**

***Index Terms*—multimedia; retrieval; ranking; unsupervised; manifold; hypergraph**

I. Introduction

MAINLY due to the universal popularity of mobile devices embedded with cameras, several users' habits have changed profoundly. Due to the advances in multimedia acquisition, storage, and sharing, billions of people were projected to the Web sharing and browsing multimedia content. In this scenario, Multimedia Information Retrieval (MIR) systems, which make use of the representation of visual content to search and retrieve relevant multimedia data, have attracted increasing attention from industry and academy [1], where various relevant benchmarks have been established [2].

In many computer vision and machine learning applications, content-based representations of multimedia data are commonly modeled as high-dimensional points in a feature space. Therefore, similarity measurement is an essential component in search and learning algorithms, once effective results depend critically on a good metric over their input space [3]–[5]. In addition, a similarity (or distance) measure is commonly used as basis for performing ranking tasks in retrieval systems.

Due to the relevance of this topic, many different approaches [4], [6]–[12] have been employed for metric learning, exploiting supervised, semi-supervised, and even unsupervised learning paradigms. In this paper, we focus on unsupervised approaches, whose objective consists in learning more effective measures or re-ranking the initial retrieval results without any labeled data or user intervention. Unlike traditional pairwise measures (e.g., those based on the Euclidean distance), such approaches compute more global similarity/distance measures capable of taking into account the information about relationships among data samples encoded in the dataset.

A myriad of unsupervised methods has been proposed, ranging from traditional image retrieval to more sophisticated person re-identification systems [6]–[9], [13]. One important class of methods relies on diffusion processes [5], [14]–[17], which have been established as a traditional research venue to compute more global contextual measures. Such methods use a pairwise affinity matrix as input, interpreted as a graph that encodes similarity information from the dataset. The pairwise affinities are re-evaluated in the context of all other elements, by diffusing similarity values through the graph [16]. Recently, re-ranking and rank-based methods [9], [18]–[20] have also attracted a lot of attention. In general, the reasoning behind such methods consists in analyzing information encoded in ranked lists in order to compute more effective ranking functions.

Manifold learning methods have also been investigated for retrieval and ranking tasks [21], [22]. The objective of such methods is to rank collection objects by taking into account the intrinsic global manifold structure, collectively revealed by the dataset being considered. The high-dimensional points, which represent multimedia content, are often located in a set of low-dimensional manifolds, which in turn can be used for computing similarity scores. Based on similarity information provided by pairwise measures or ranking information, such methods learn more global affinity measures capable of considering the intrinsic structure of the dataset manifold. Manifold learning methods have been established as relevant research topic for years, especially more recently [10]–[12], [21]–[24].

In this paper, we propose a novel unsupervised manifold learning algorithm for multimedia retrieval and ranking tasks, called *Log-based Hypergraph of Ranking References* (LHRR). The proposed hypergraph representation and the respective hyperedges are based on Ranking References, to which are assigned weights according to a log-based function. The hypergraphs are a powerful generalization of graphs, which allow to define hyperedges capable of connecting any number of vertices and representing similarity relationships among sets of objects, instead of only pairs, as in traditional graph-based approaches. Identifying a set of similar objects is of crucial relevance for capturing the dataset manifold structure and effective ranking.

Hypergraphs have been widely exploited in multimedia retrieval approaches with positive results, including unsuper-

vised and especially semi-supervised scenarios. Some representative works include, for example [25]–[28]. However, in unsupervised image retrieval scenarios where re-ranking methods take into account the intrinsic structure of the dataset manifold for learning more effective measures, only few and recent methods have been proposed based on hypergraphs [15].

Another research problem addressed in this paper consists in the development of fusion approaches [29]. Diverse feature representations often contribute to a better similarity measure. Multimedia data are usually described by multiple features associated with multiple views, which often provide complementary information about their content [4]. Both rank [7], [20] and diffusion [4] based approaches have been exploited for fusion tasks, indicating the relevance of combining different views to reach more effective results. In this paper, we also validate the proposed method in this context, showing how it can be used not only for re-ranking a single feature, but also for combining different ranking inputs.

In summary, the main contributions of this paper are:

- Despite the intense use of hypergraphs on vision, learning and manifold ranking tasks [30]–[33], including unsupervised [34]–[36] and semi-supervised scenarios [25], [26], [28], few works [15] have recently exploited them for unsupervised re-ranking on image retrieval. The main novelty of the proposed LHRR approach is a rank-based model proposed for the hypergraph construction. The definition of hyperedges, including input/output data, are completely based on unsupervised ranking information, weighted by a log function.
- The rank-based model makes the method independent of distance measures and enables efficient algorithmic solutions, as discussed in this paper;
- The dataset manifold structure is captured through a hypergraph-based similarity measure. The proposed approach exploits the hypergraph structure through complementary aspects, improving the produced ranking results.

A broadly and extensive experimental evaluation was conducted considering several different retrieval scenarios. The evaluation was conducted on nine public datasets, including seven image datasets and two video datasets. Several different image and video features were considered, including global (shape, color, and texture), local, deep-learning- and motion-based descriptors. The proposed method achieved very significant effectiveness gains, reaching up to +109% of relative gain on certain datasets. Comparisons with other recent methods on different datasets were also conducted and the proposed LHRR algorithm yielded very high effectiveness performance in comparison with various state-of-the-art approaches.

The remainder of this paper is organized as follows. Section II discusses related work and Section III formally describes the ranking problem addressed. Section IV presents the proposed *Log-based Hypergraph of Ranking References* (LHRR) method and Section V discusses an efficient algorithmic solution for computing the method. Section VI describes the conducted experimental evaluation and, finally, Section VII concludes this paper and provides possible future research directions.

II. Related Work

Hypergraphs are a generalization of graphs and, although not so broadly used as simple graphs, have been attracted attention in the last decades [37]. While graphs often model pairwise relationships, in many real-world problems, relationships among objects are more complex than pairwise. In this scenario, hypergraphs allow capturing high-order relations in various domains [38], [39]. Due to the significant developments of combinatorics in conjunction with computer science, hypergraphs are nowadays increasingly relevant in science and engineering applications [37].

In learning applications hypergraphs were first used on clustering algorithms. In [40], a hypergraph is approximated through a weighted graph. Then, a spectral partitioning algorithm is used to partition the vertices of this graph. In [38], spectral clustering was generalized to operate on hypergraphs, also supporting other tasks as hypergraph embedding and transductive classification.

In last decade, the use of hypergraph on computer vision and machine learning areas has spread to significant number of applications. A hypergraph spectral learning formulation was proposed for multi-label classification in [39]. A probabilistic hypergrah ranking was proposed in [27], [28], considering a semi-supervised learning scenario. In fact, the applicability of hypergraph on semi-supervised learning scenarios is remarkable [25]–[28], [30], [41]. Semi-supervised learning algorithms based on hypergraphs have been used for text summarization [41], image classificaton [26], and visual search re-ranking [25]. More recently, Lit et al. [30] used a hypergraph Laplacian matrix for deriving an algorithm for clustering and semi-supervised classification. Mainly due to its versatility and capacity of modeling high-order relationships, various other hypergraph applications have been established last years, as: person re-identification [42], multimodal retrieval and re-ranking [31], [43], gait recognition [44] and feature selection [32], [45]. Studies regarding specific properties of hypergraphs have been also recently conducted [46].

In another promising research direction, manifold and metric learning approaches also have been attracting a lot of attention, specially recently [4], [10]–[13], [23], [24], [47]–[49]. The capacity of exploiting the intrinsic global manifold structure of datasets represent a significant advantage in diverse ranking and learning tasks. In [11], an algorithm is proposed to capture the image manifold in the feature space through a regional diffusion mechanism. In [21], the similarity measured on a manifold is estimated by a random walk process. A spectral ranking is proposed in [22], which uses a an explicit embedding for reducing the manifold search to a two-stage similarity search. Recently, a manifold learning algorithm was also used to incorporate the global topological structure of dataset into learning hashing function procedure [12].

In general, metrics are the basis of many learning algorithms, and their effectiveness often presents significant impact on results [49]. In [47], a multi-manifold metric learning is proposed for deep representations. In [4], metric fusion is conducted over multiview data through a graph random walk algorithm. It can be observed that graphs are tools

commonly used for modeling various manifold and metric learning algorithms [4], [10], [24], [50].

Hypergraph models have also been exploited in unsupervised retrieval scenarios. The initiative of Zhu et al. [34], for example, proposed to use an unsupervised hypergraph representation to support effective and efficient mobile image retrieval. The novelty of their work was on the encoding of both image content and associated texts into a representation, which can later be used to generate discriminating binary codes. The goal is to improve image search tasks by taking advantage of high-order semantic correlations of images. The work of Gao et al. [35], in turn, addressed the problem of 3-D object retrieval and recognition. In their work, multiple hypergraphs were used to represent 3-D objects based on their 2-D views, where each vertex is an object and an edge encodes a cluster of views. Their goal is to avoid the direct computation of a distance among objects. In the work of Huang [36], hypergraph models are exploited in two scenarios: (*i*) video object segmentations and (*ii*) content based image retrieval. The first application uses an unsupervised hypergraph cut algorithm for clustering, which involves eigen-decomposition of the hypergraph Laplacian matrix. The second discusses a semi-supervised learning algorithm based on a probabilistic hypergraph, which involves the solving of a linear system. The main focus is on image retrieval based on relevance feedback.

Despite the above related works addressed the use of hypergraphs in image retrieval tasks in an unsupervised fashion, none of them focused on the re-ranking problem, the goal of our paper. In a recent work [15], the most related to ours, a regularized diffusion process is discussed in unsupervised object retrieval scenarios, deriving a generic tool for tensor-order affinity learning among objects. Hypergraphs are used for 3D models grouped into multiple clusters, where each cluster is deemed as one hyperedge. In contrast, our proposed hypergraph model is derived completely based on ranking references and does not require any clustering step. In spite of that, the proposed method is capable of capturing the intrinsic structure of datasets and exploiting the powerful of hypergraphs on representing high-order similarity relationships for ranking tasks.

III. Problem Formulation

This section discusses the problem formulation and the notation used through the paper.

A. Feature Extraction and Similarity Computing

The object content is firstly encoded through a feature extraction procedure, which allows quantifying the similarity between multimedia objects (images, videos). Let $\mathcal{D}$ be a descriptor, which can be defined as a tuple (ϵ, δ), where:

- ϵ: $o_i \rightarrow \mathbb{R}^d$ is a function, which extracts a feature vector $v_{\vec{i}}$ from a multimedia object o_i;
- δ: $\mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}^+$ is a function that computes the distance between two multimedia objects according to the distance between their corresponding feature vectors.

The distance between two objects o_i, o_j is computed as $\delta(\epsilon(o_i), \epsilon(o_j))$. The Euclidean distance is commonly used to compute δ, although the proposed ranking method is independent of distance measures. A similarity measure $\rho(o_i, o_j)$ can be computed based on distance function δ and used for ranking tasks. The notation $\rho(i, j)$ is used along the paper.

B. Multimedia Retrieval and Rank Model

Let $\mathcal{C}$=$\{o_1, o_2, \ldots, o_n\}$ be a multimedia collection, where $n = |\mathcal{C}|$ denotes the size of the collection $\mathcal{C}$. The target task refers to retrieving multimedia objects (images, videos) from $|\mathcal{C}|$ based on their content. Let o_q denote a query object. A ranked list τ_q can be computed in response to o_q based on the similarity function ρ. The ranked list τ_q=$(o_1, o_2, \ldots, o_n)$ can be defined as a permutation of the collection $\mathcal{C}$. A permutation τ_q is a bijection from the set $\mathcal{C}$ onto the set $[N] = \{1, 2, \ldots, n\}$. For a permutation τ_q, we interpret $\tau_q(i)$ as the position (or rank) of the object o_i in the ranked list τ_q. If o_i is ranked before o_j in the ranked list of o_q, i.e., $\tau_q(i) < \tau_q(j)$, then $\rho(q, i) \geq \rho(q, j)$.

The top positions of ranked lists are expected to contain the most similar objects to the query object. Additionally, τ_q can be expensive to compute, specially when n is high. Therefore, the computed ranked lists can consider only a sub-set of the collection. Let τ_q be a ranked list that contains only the L most similar objects to o_q, where $L \ll n$. Formally, let $\mathcal{C}_L$ be a sub-set of the collection $\mathcal{C}$, such that $\mathcal{C}_L \subset \mathcal{C}$ and $|\mathcal{C}_L| = L$. The ranked list τ_q can be defined as a bijection from the set $\mathcal{C}_L$ onto the set $[N] = \{1, 2, \ldots, L\}$. Every object $o_i \in \mathcal{C}$ can be taken as a query o_q. A set of ranked lists $\mathcal{T}$ = $\{\tau_1, \tau_2, \ldots, \tau_n\}$ can also be obtained, with a ranked list for each object in the collection $\mathcal{C}$.

Based on the rank model, the neighborhood set can also be defined. Let o_q be a multimedia object taken as query, a neighborhood set $\mathcal{N}(q, k)$ that contains the k most similar multimedia objects to o_q can be defined as follows:

$$\mathcal{N}(q,k) = \{\mathcal{S} \subseteq \mathcal{C}, |\mathcal{S}| = k \wedge \forall o_i \in \mathcal{S}, o_j \in \mathcal{C} - \mathcal{S} : \tau_q(i) < \tau_q(j)\}. \quad (1)$$

C. Manifold Ranking Formulation

The main objective of the proposed LHRR method is to exploit the similarity information encoded in the set of ranked lists $\mathcal{T}$, capturing the structure of the dataset manifold. Based on such analysis, a new and more effective set of ranked $\mathcal{T}_r$ is computed in an unsupervised way with the aim of improving the effectiveness of retrieval tasks. More formally, we can describe the method as function f_r, such that $\mathcal{T}_r = f_r(\mathcal{T})$.

Additionally, the fusion problem is also considered, in which different sets of ranked lists $\{\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_d\}$ are taken as input aiming at computing a more effective set $\mathcal{T}_r$.

IV. Hypergraph Manifold Ranking

The proposed method aims at performing context-aware ranking tasks by capturing the dataset manifold struture and identifying global similarity relationships. The algorithm exploits the hypergraph ability of representing high-order relationships in order to model expanded similarity connections. Actually, representing only pairwise relations through simple graphs it is not complete for many multimedia retrieval tasks, where modelling relationships among set of objects can be

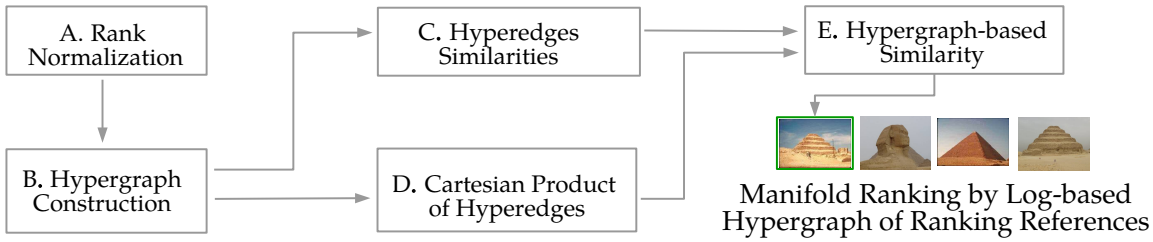


Fig. 1: The main steps of the proposed *Log-based Hypergraph of Ranking References*: the ranked lists are normalized for improved symmetry (A) and used for the hypergraph model constructed (B). Information from both the similarity between hyperedges (C) and the Cartesian product of hyperedges (D) are combined to defined a more effective similarity measure (E).

of paramount importance [27]. Through hypergraphs, vertices with similar characteristics can all be enclosed by a hyperedge, and thus high-order information can be properly captured [30].

Different from most of hypergraph approaches, the assignment of vertices to hyperedges is not binary. Although weighted hypergrahs have already been exploited [12], [27], a novel approach is proposed in this paper to estimate if a vertex belongs to a hyperedge, based only on the input rank information. In this way, the method is completely independent of feature extraction procedures, mid-level representations, and distance measures.

The proposed *Log-based Hypergraph of Ranking References* (LHRR) is composed of five main steps, whose objective is to encode the similarity of objects from different perspectives. Figure 1 illustrates the overall organization of the proposed method in terms of its main components, and existing dependencies. Each of the main steps of the algorithm is briefly described in the following and formally defined in the next sub-sections.

A. **Rank Normalization:** a normalization procedure is performed to improve the symmetry of ranking references;
B. **Hypergraph Construction:** the hypergraph models the global similarity structure of the dataset using the rank information as input;
C. **Hyperedge Similarities:** the relationships encoded in the hyperedges are used to computed a novel similarity between multimedia objects;
D. **Cartesian Product of Hyperedge Elements:** a Cartesian product operation is computed for maximizing similarity information from hyperedges;
E. **Hypergraph-Based Similarity:** the similarities between hyperedges and the Cartesian product operations are combined to compute a hypergraph-based similarity, which leads to new rankings.

A. Rank Normalization

Different from most of pairwise distance (or similarity) measures, the relationships established by ranking references and k-neighborhood sets are not symmetric. The benefits of improving the symmetry of the k-neighborhood relationships are remarkable in image retrieval tasks [10], [51], specially when ranking information is used to model the similarity structure of datasets.

A simple reciprocal rank normalization is considered, computing a new similarity measure ρ_n based on the reciprocal rank positions:

$$\rho_n(i,j) = 2L - (\tau_i(j) + \tau_j(i)) \tag{2}$$

Based on the computed measure, the multimedia objects at the top-L positions of the ranked lists are updated by a stable sorting algorithm.

B. Hypergraph Construction

A hypergraph is a generalization of the traditional graph, in which the edges are non-empty subsets of the vertex set and therefore can connect any number of vertices [27], [37]–[39]. Given a hypergraph $G = (V, E, w)$, the set V represents a finite set of vertices and E denotes the hyperedge set. The set of hyperedges E can be defined as a family of subsets of V such that $\bigcup_{e \in E} = V$. Each vertice $v_i \in V$ is associated with an object $o_i \in \mathcal{C}$. To each hyperedge e_i, a positive weight $w(e_i)$ is assigned, which denotes the confidence of relationships established by the hyperedge e_i.

A hyperedge e_i is said to be incident with a vertex v_j when $v_j \in e_i$. In this way, a hypergraph can be represented by an incidence matrix $\mathbf{H}_b$ of size $|E| \times |V|$:

$$h_b(e_i, v_j) = \begin{cases} 1, & \text{if } v_j \in e_i, \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

A hyperedge e_i can be defined as a set of vertices $e_i = \{v_1, v_2, \ldots, v_m\}$. Therefore, the matrix $\mathbf{H}_b$ allows only a binary assignment of a vertex to a hyperedge, while in many situations, it is desired to consider a degree of uncertain. In order to overcome this limitation, probabilistic hypergraphs have been exploited [27], representing also the probability that a vertex belongs to a hyperedge. Let $r : E \times V \rightarrow \mathbb{R}^+$ be a function with a codomain in the $\mathbb{R}^+$, a continuous incidence matrix $\mathbf{H}$ can be defined as:

$$h(e_i, v_j) = \begin{cases} r(e_i, v_j), & \text{if } v_j \in e_i, \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

*1) **Hyperedge Definition**:* A hyperedge e_i is defined for each object $o_i \in \mathcal{C}$ based on the k-neighborhood set of o_i and its respective neighbors. Let $o_x \in \mathcal{N}(i,k)$ be a neighbor of o_i and let $o_j \in \mathcal{N}(x,k)$ be a neighbor of o_x. The membership measure $r(e_i, v_j)$, which indicates the degree to which the vertex v_j belong to a hyperedge e_i, is computed as:

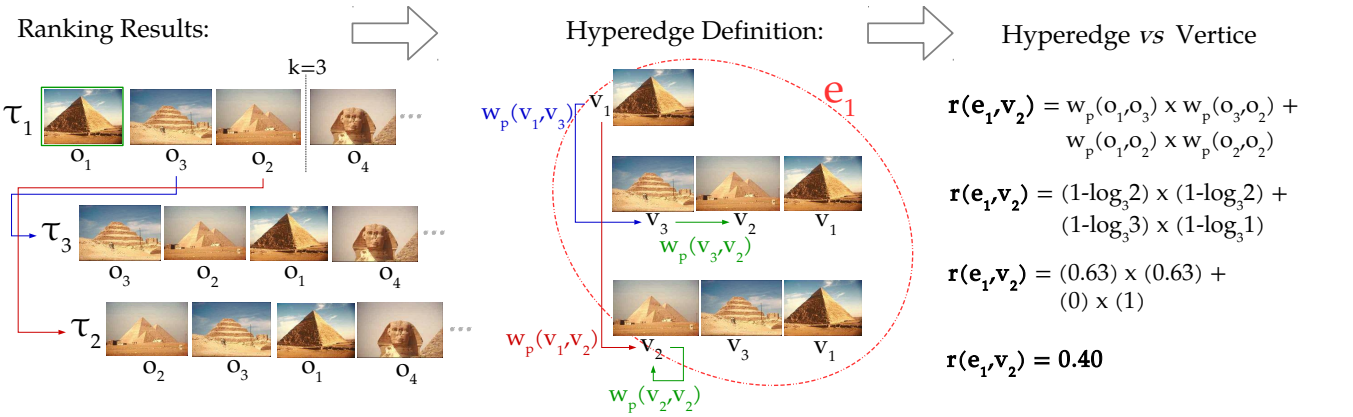
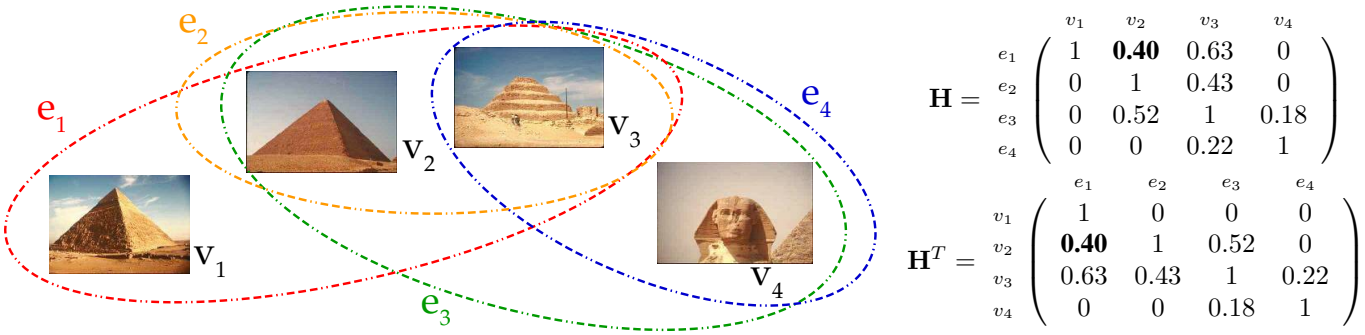


Fig. 2: Illustration of a hyperedge definition (e_1) based on Ranking References with a neighborhood size of k=3. The function w_p assigns weights according to positions and is used to define the association between a hyperedge and a vertice ($r(e_1, v_2)$).



$$\mathbf{H} = \begin{array}{c} \\ e_1 \\ e_2 \\ e_3 \\ e_4 \end{array} \begin{array}{c} \begin{array}{cccc} v_1 & v_2 & v_3 & v_4 \end{array} \\ \begin{pmatrix} 1 & \mathbf{0.40} & 0.63 & 0 \\ 0 & 1 & 0.43 & 0 \\ 0 & 0.52 & 1 & 0.18 \\ 0 & 0 & 0.22 & 1 \end{pmatrix} \end{array}$$

$$\mathbf{H}^T = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \end{array} \begin{array}{c} \begin{array}{cccc} e_1 & e_2 & e_3 & e_4 \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 0 \\ \mathbf{0.40} & 1 & 0.52 & 0 \\ 0.63 & 0.43 & 1 & 0.22 \\ 0 & 0 & 0.18 & 1 \end{pmatrix} \end{array}$$

Fig. 3: Similarity among images modelled by LHRR: vertices $\{v_1,v_2,v_3,v_4\}$ and hyperedges $\{e_1,e_2,e_3,e_4\}$. Hyperedges represent the similarity relationships among sets of images: $e_1(v_1, v_2, v_3)$, $e_2(v_2, v_3)$, $e_3(v_2, v_3, v_4)$, and $e_4(v_3, v_4)$. On the right, the respective incidence matrix $\mathbf{H}$ and the transposed matrix $\mathbf{H}^T$. The hyperedge e_1 follows the example illustrated in Figure 2.

$$r(e_i, v_j) = \sum_{o_x \in \mathcal{N}(i,k) \wedge o_j \in \mathcal{N}(x,k)} w_p(i, x) \times w_p(x, j), \quad (5)$$

where $w_p(i, x)$ is a function that assigns a weight of relevance to o_x according to its position in the ranked list τ_i. Objects which are close to each other in the feature space are expected to be relevant to each other, i.e., are expected to belong to same "semantic" class. This is the rationale behind the hyperedge definition.

The size of the hyperedge $|e_i|$ can vary according to the number of elements in common in the top-k positions τ_i at its respective k-neighbors. More specific, the size varies from k (when all elements are the same) to $(k^2 - k)$ (when all the elements are different). A high diversity of elements may indicate a high degree of uncertainty and this information will be exploited for defining the weights of hyperedges in the next sub-section.

The weight assigned to o_x according to its position in the ranked list τ_i is defined as follows:

$$w_p(i, x) = 1 - \log_k \tau_i(x). \quad (6)$$

The function $w_p(i, x)$ assigns a maximum weight of 1 to the first position (the query), presenting a quickly decay for the first rank positions. The objective is to assign high weight to the top positions, where the effectiveness of ranked lists is superior. Notice that our method uses only ranking information for defining the hypergraph and respective hyperedges, differing from other probabilistic hypergraph approaches [27].

Figure 2 illustrate how the ranking references were exploited to define a hyperedge (e_1) and how is computed the relationship among the hyperedge and a vertice. The hyperedge e_1 is defined based on the ranked list τ_1 and its respective references to τ_2 and τ_3. The function w_p is represented for defining the weights for each vertice/object according to its position in the ranked list. The association between the hyperedge e_1 and the vertice v_2 is defined by $r(e_1, v_2)$, which is computed based on reference weights given by w_p.

Figure 3 illustrates an example of a whole hypergraph with four vertices and hyperedges. The respective incidence matrix and its transposed (which will be used in next steps) are also represented. The hyperedge e_1, used as example in Figure 2, is part of the hypergraph illustrated in Figure 3. The values in the matrix $\mathbf{H}$ are computed as showed on the right part of Figure 2.

*2) **Hyperedge Weight**:* The weight of a hyperedge $w(e_i)$ denotes the confidence of relationships established among vertices by the hyperedge. As previously mentioned, the lower the number and diversity of vertices in the hyperedge, the higher tends to be the quality of hyperedge and the values of

$h(e_i, \cdot)$.

In order to compute the weight $w(e_i)$, we first define the Hypergraph Neighborhood Set $\mathcal{N}_h$. Given an hyperedge e_i, a set $\mathcal{N}_h$ with the vertices with the greatest $h(e_i, \cdot)$ are formally defined as:

$$\mathcal{N}_h(q,k) = \{\mathcal{S} \subseteq e_q, |\mathcal{S}| = k \wedge \forall o_i \in \mathcal{S}, o_j \in e_q - \mathcal{S} : h(q,i) > h(q,j)\}. \quad (7)$$

A high-effective hyperedge is expected to contain a few vertices, which therefore are related to high values of $h(e_i, \cdot)$. In this way, the weight $w(e_i)$ is defined as:

$$w(e_i) = \sum_{j \in \mathcal{N}_h(i,k)} h(i,j). \quad (8)$$

Once the hyperedge is defined based on the ranked list τ_i, the weight $w(e_i)$ can also be seen as an unsupervised effectiveness estimation of the ranked list τ_i. Therefore, high values of $w(e_i)$ are assigned to high-effective ranked lists.

C. Hyperedge Similarities

While the hypergraph is a powerful model to represent high-order similarity relationships, in certain circumstances it is necessary to extract the similarity information in pairwise form, for example to perform ranking tasks. Other works [40] have addressed the problem of approximating the hypergraph with a graph. In this paper, we propose a novel approach for exploiting the hyperedge similarities in order to compute a pairwise similarity matrix $\mathbf{S}$. The pairwise similarity is computed based on two different views which are combined.

The first hypothesis is that similar objects present similar ranked lists and, therefore, similar hyperedges. Once all similarity information is encoded by the incidence matrix H, a similarity measure between two hyperedges e_i, e_j can be computed by a sum of of h values multiplied on the correspondent vertices: $h(e_i, v_x) \times h(e_j, v_x)$. Such operation can be modeled for all elements by multiplying the incidence matrix and its transposed, as follows:

$$\mathbf{S}_h = \mathbf{H}\mathbf{H}^T \quad (9)$$

The second hypothesis states that similar objects are expected to be referenced by the same hyperdges. In this way, to compute a pairwise similarity between two vertices v_i, v_j, the h values on correspondent hyperedges should be multiplied: $h(e_x, v_i) \times h(e_x, v_j)$. The operation can be computed by multiplying transposed incidence matrix $\mathbf{H}^T$, as follows:

$$\mathbf{S}_v = \mathbf{H}^T\mathbf{H} \quad (10)$$

Since both similarities between vertices and hyperedges encoded relevant and complementary information, they are combined through a multiplication element by element $s(i,j) = s_h(i,j) \times s_v(i,j)$. Therefore, the pairwise similarity matrix can be computed by a Hadamard product:

$$\mathbf{S} = \mathbf{S}_h \circ \mathbf{S}_v \quad (11)$$

Notice that all matrices considered in this section are very sparse. Consequently, an efficient algorithmic solution can be derived to compute $s(i,j)$, as further discussed in Section V.

D. Cartesian Product of Hyperedge Elements

As previously discussed, each hyperedge connects a set of vertices. In order to extract pairwise relationships direct from the set of elements defined by a hyperedges, a Cartesian product operation is conducted. The objective is to maximize similarity information, which can be aggregated to the hyperedge similarities. Given two hyperedges $e_q, e_i \in E$, the Cartesian product between them can be defined as:

$$e_q \times e_i = \{(v_x, v_y) : \ v_x \in e_q \wedge v_y \in e_i\}. \quad (12)$$

Let ${e_q}^2$ denote the Cartesian product between the elements of the same hyperedge e_q, such that $e_q \times e_q = {e_q}^2$. For each pair of vertices $(v_i, v_j) \in {e_q}^2$ a pairwise similarity relationship $p : E \times V \times V \rightarrow \mathbb{R}^+$ is established. The function p is computed based on the weight $w(e_q)$, which denotes the confidence of hyperedge that originates the association. The membership degrees of v_i and v_j are defined as:

$$p(e_q, v_i, v_j) = w(e_q) \times h(e_q, v_i) \times h(e_q, v_j). \quad (13)$$

A similarity measure based on Cartesian product is defined through a matrix $\mathbf{C}$, which considers relationships contained in all the hyperedges. The reasoning behind this formulation relies on taking into account the co-occurrence of v_i and v_j in different hyperedges, accumulating its respective $p(\cdot, v_i, v_j)$ values. Each position of the matrix $\mathbf{C}$ is computed as follows:

$$c(i,j) = \sum_{e_q \in E \wedge (v_i, v_j) \in {e_q}^2} p(v_i, v_j) \quad (14)$$

E. Hypergraph-Based Similarity

The pairwise similarity defined based on hyperedges and Cartesian product operations provide distinct and complementary information about the dataset manifold. Therefore, both information is exploited by computing an affinity matrix $\mathbf{W}$ which combines $\mathbf{C}$ and $\mathbf{S}$ as follows:

$$\mathbf{W} = \mathbf{C} \circ \mathbf{S} \quad (15)$$

Based on the affinity measure defined by $\mathbf{W}$, a ranking procedure can be performed given rise to a new set of ranked lists $\mathcal{T}$. Once both the input and output of the method is defined in terms of ranking information, the process can be iteratively repeated. Let the superscript $^{(t)}$ denote the current iteration and let $\mathcal{T}^{(0)}$ denote the set of ranked lists defined by the feature, we can say that $\mathcal{T}^{(t+1)}$ can be computed based on $\mathbf{W}^{(t)}$. After a certain number of T iterations a final set of ranked lists $\mathcal{T}_r$ is obtained.

F. Hypergraph Manifold Rank Fusion

Defining a broad and complete representation for multimedia content is a very difficult task. Visual information, for example, is composed of many distinct aspects, which can be hard to encode in single feature. In this scenario, fusion approaches have been seen as a promising research direction [4], [20], [29]. Diverse feature representations can be exploited to overcome this limitation, since as visual content

described by multiple features can be decomposed into multiple views, thus often providing distinct and complementary information [4]. The diversity and complementarity offered by different features can substantially improve the effectiveness of retrieval tasks. In this paper, we propose to exploit the capacity of LHRR algorithm in capturing the dataset manifold to combine distinct input rankings computed by different features. We also addressed a remarkable challenge in fusion tasks, which consists in defining adaptive weights for each feature [52].

Let $\{\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_m\}$ denote a set, in which each element $\mathcal{T}_d$ is a set of ranked lists computed by a feature d. Since the most significant effectiveness gains are obtained at the first iteration, the LHRR algorithm is computed independently for each feature, considering one iteration ($T = 1$). In this way, a set of ranked lists $\mathcal{T}_d^{(1)}$ is computed for each feature. Next, a rank-based formulation is used to combine the ranked lists, exploiting an adaptive weight, which is assigned to each query/feature according to the weight of the respective hyperedge. Let w_f denote the fused affinity measure, each element is computed as follows considering the top-L positions of τ_q:

$$w_f(q,i) = \prod_{d=1}^{m} \frac{(1 + w_d(e_q))}{(1 + \log_L \tau_{q,d}(i))}, \tag{16}$$

where $w_d(e_q)$ is the weight of hyperedge e_q according to the feature d and $\tau_{q,d}(i)$ denote the position of o_i in the ranked list of o_q according to the feature d. The combined affinity measure $w_f(\cdot,\cdot)$ gives rise to a unique set of ranked lists $\mathcal{T}_f$. This set is then processed by the LHRR algorithm as a single feature and modeled by a unified hypergraph.

V. Efficient Algorithimic Solution

Recently, in addition to effectiveness aspects, efficiency and scalability properties of unsupervised post-processing methods have also been attracted a lot of attention [18], [53]. Due to complex operations required by most of diffusion-based approaches, the use of some approaches are computationally prohibited for larger datasets [18]. In this scenario, we discuss in this section an efficient algorithmic solution for computing the LHRR method.

In the way it was defined in previous section, the complexity of the proposed method is delimited as $O(n^3)$ by the matrix multiplication operations given by Equations 9 and 10. However, the matrix $\mathbf{H}$ is extremely sparse, with at most $(k^2 - k)$ elements filled at each line. Therefore, a sparse matrix and an adjacency list can be maintained to compute each element in $O(k^2)$, reducing the overall complexity to $O(n^2)$. Additionally, since the method is completely based on ranking information, it is not required that the resulting matrix is fully computed. In fact, a sparse matrix can be computed considering only the top-L positions of ranked lists, where L is constant significantly smaller than n, specially for large datasets. Since it is highly unlikely that relevant objects are found after top-L positions, a very efficient algorithm can be designed without significant lost in effectiveness (as discussed in Section VI-B).

Based on these assumptions, Algorithms 1 and 2 present efficient $O(n)$ solutions for computing matrices $\mathbf{S}_h$ and $\mathbf{S}_v$ equivalent to the Equations 9 and 10, respectively. Notice that loops defined on lines 2 and 3 of Algorithms 1 are independent of the collection size. The same can be said about loops of lines 2, 7, and 8 of Algorithm 2.

The Cartesian product operations, which also constitute a relevant step of the algorithm is less expansive computationally and can be efficiently computed in $O(n)$. A set of increments are computed based on Cartesian product of hyperedges to compute a sparse matrix $\mathbf{C}$. Algorithm 3 presents such algorithmic solution. Loops of lines 2 and 3 define the Cartesian product for each hyperedge.

Other associated steps of the algorithm can also be efficiently computed in $O(n)$ based on the conjecture that only objects at top-L positions of ranked lists should processed. The rank normalization (Equation 2), the Hadmard product (Equations 11 and 15) and the re-sorting of ranked lists computed for the top-L positions have complexity of $O(nL)$. The re-sorting step uses the insertion sort algorithm, which tends to present complexity of $O(L)$ for ranked lists almost sorted. Therefore, assuming L constant, the overall complexity of the LHRR method is $O(n)$. The implementation of the proposed the algorithm is public available under the UDLF framework [54].

Algorithm 1 Hyperedge Similarities Computing.

Require: Set of ranked lists $\mathcal{T}$ and set of hyperedges E
Ensure: Sparse similarity matrix $\mathbf{S}_h$
1: **for all** $o_i \in \mathcal{C}$ **do**
2: **for all** $o_j \in \mathcal{N}(i, L)$ **do**
3: **for all** $v_x \in e_i$ **do**
4: **if** $v_x \in e_j$ **then**
5: $s_h(i,j) \leftarrow s_h(i,j) + (h(i,x) \times h(j,x))$
6: **end if**
7: **end for**
8: **end for**
9: **end for**

Algorithm 2 Similarity among vertices references.

Require: Set of ranked lists $\mathcal{T}$ and set of hyperedges E
Ensure: Sparse similarity matrix $\mathbf{S}_v$
1: **for all** $e_i \in E$ **do**
2: **for all** $v_j \in e_i$ **do**
3: $r(v_j) = r(v_j) \cup e_i$
4: **end for**
5: **end for**
6: **for all** $o_i \in \mathcal{C}$ **do**
7: **for all** $o_j \in \mathcal{N}(i, L)$ **do**
8: **for all** $e_x \in r(v_i)$ **do**
9: **if** $e_x \in r(v_j)$ **then**
10: $s_v(i,j) \leftarrow s_v(i,j) + (h(e_x, v_i) \times h(e_x, v_j))$
11: **end if**
12: **end for**
13: **end for**
14: **end for**

Algorithm 3 Similarity based on Cartesian product operations.

Require: Set of ranked lists $\mathcal{T}$ and set of hyperedges E
Ensure: Sparse Cartesian product matrix $\mathbf{C}$
for all $o_q \in \mathcal{C}$ **do**
  for all $o_i \in e_q$ **do**
    for all $o_j \in e_q$ **do**
      $c(i,j) \leftarrow c(i,j) + (w(e_q) \times h(e_q, v_i) \times h(e_q, v_j))$
    end for
  end for
end for

VI. Experimental Evaluation

The proposed method was evaluated through a rigorous and extensive experimental evaluation, considering various and diversified datasets and features through different retrieval tasks. Section VI-A describes the datasets, features, and the protocol adopted in the experimental evaluation. Section VI-B discusses the impact of parameters. Section VI-C presents the experimental results obtained on shape, color, and texture retrieval tasks. Experiments involving natural image retrieval and object retrieval tasks are discussed, respectively, in Sections VI-D and VI-E. Video retrieval tasks are discussed in Section VI-F. Different aspects of the method are analyzed in Section VI-G while Section VI-H presents qualitative and visual results. Finally, a comparison with other state-of-the-art methods is presented in Section VI-I.

A. Datasets, Features and Experimental Protocol

The LHRR method was extensively evaluated on seven well-known public image datasets and two video datasets. A diverse range of datasets are considered, including varied sizes of datasets and images/videdos with diversified characteristics, ranging from 280 images to 87,648 multimedia videos. Several different features are used, global (shape, color, and texture properties), local, mid-level representations and convolutional neural network-based features. Multifaceted conditions were considered to evaluate the robustness of the proposed LHRR method in retrieval tasks. Table I describes the main characteristics of the datasets and the features used for each dataset.

All images are considered as query images except for the Holidays [66] dataset, for which we use 500 queries due to comparison purposes. The effectiveness measure considered for most of experiments is the Mean Average Precision (MAP), but other measures are also considered according to the specific protocol of some datasets: the N-S score [79] is used for UKBench [79] dataset and the Recall at 40 (bull's eye score) for MPEG-7 [59] dataset. The Precision (P@x) is also used in some analysis. Most of experiments also report the relative gains obtained by the LHRR method, which is defined as follows: let M_b, M_a be the effectiveness measure respectively before and after the use of the LHRR, the relative gain is defined as $G = \frac{M_a - M_b}{M_b}$.

B. Impact of Parameters

Only two parameters are required by the LHRR method: k, which denotes the neighborhood size and T, which defines the number of iterations. The method also considers the value of L, which denotes the size of ranked list considered, defining a trade-off between effectiveness and efficiency.

Firstly, an experiment was conducted for a jointly analysis of the two parameters. The values k and T are varied and the impact on the MAP measure is evaluated. Figure 4 presents the obtained results. As we can observe, a smooth surface with a large red region was obtained for both datasets, indicating the robustness of the method for different parameter settings. In fact, the method converges very quickly as further discussed in Section VI-G. Therefore, the value of $T = 2$ were used on all datasets. Fusion tasks used $T = 1$.

The impact of neighborhood size k is also analyzed individually on different datasets. Figure 5 present the results, considering various distinct features. A large increase of effectiveness can be observed for small k values until reaching an stabilization. Regarding the neighborhood size, most of experiments report the effectiveness scores in two scenarios: using fixed parameters values and using the best k parameter. The objective is to report the highest gains and, at same time, to evaluate the method in general situations. The fixed neighborhood size k is defined as 5 for instance retrieval datasets (Holidays [66] and UKBench [79]), 70 for video datasets and 20 for the other image datasets.

The trade-off defined by L is evaluated on two datasets of very different sizes. The results are presented in Figure 6. A small value of L in comparison with the size of the dataset is enough to reach high-effective results. Additionally, an accentuated increase of effectiveness is observed in the beginning of the curve.

C. Shape, Color, and Texture Retrieval

The LHRR is firstly evaluated in general image retrieval tasks considering shape, color, and texture properties. Table II presents the obtained results. The MAP score is reported for the original descriptor and for the LHRR method, including rank fusion tasks. The results considered a fixed k and the value of k which achieved the highest MAP score. Different values of L are also reported in order to allow the analysis of the impact on effectiveness.

The best result for each descriptor is highlighted in bold. Very expressive gains up to +44.84% can be observed. A rank fusion of CFD+ASC, for example, achieved 99.37% from initial scores of 80.71% and 85.28%. We can also notice that fixed values of k and small values of L also achieved significant gains for most of descriptors.

D. Natural Image Retrieval

The experimental evaluation considering natural image retrieval tasks were conducted on three popular datasets: the University of Kentucky Recognition Benchmark - UKBench [79], the Holidays [66] dataset, and the Corel5K [77] dataset. Since both Holidays and Corel5K are small datasets, a full value of L is used. Several different features are evaluated, including various deep-learning representations.

Table III presents the effectiveness results for the Corel5K [77] dataset. Remarkable gains can be observed for various features. We can highlight that the CNN-Caffe feature,

TABLE I: Resources considered in the experimental evaluation: image/video datasets and features used for each dataset.

Dataset	Size	Type	General Description	Descriptors	Effectiv. Measure
Soccer [55]	280	Scenes/ Color	Composed of images from 7 soccer teams, containing 40 images per class.	Border/Interior Auto Color Correlograms (ACC) [56], Pixel Classification (BIC) [57], and Global Color Histogram (GCH) [58]	MAP
MPEG-7 [59]	1,400	Images: Shape	Composed of 1400 shapes divided in 70 classes. Commonly used for evaluation of post-processing methods.	Articulation-Invariant Representation (AIR) [60], Aspect Shape Context (ASC) [61], Beam Angle Statistics (BAS) [62], Contour Features Descriptor (CFD) [63], Shape Context (IDSC) [64], and Segment Saliences (SS) [65]	MAP, Recall@40
Holidays [66]	1,491	Scenes	Commonly used as image retrieval benchmark, the dataset is composed of 1,491 personal holiday pictures with 500 queries.	Joint Composite Descriptor (JCD) [67], Scalable Color Descriptor (SCD) [68] Color and Edge Directivity Descriptor Spatial Pyramid (CEED-Spy) [69], [70], ACC [56], Convolutional Neural Network by Caffe [71] (CNN-Caffe), and Convolutional Neural Network by OverFeat [72] (CNN-OverFeat)	MAP
Brodatz [73]	1,776	Images: Texture	A popular dataset composed of 111 different textures divided into 16 blocks.	Color Co-Occurrence Matrix (CCOM) [74], Local Activity Spectrum (LAS) [75], and Local Binary Patterns (LBP) [76]	MAP
Corel5K [77]	5,000	Objects/ Scenes	Composed of 50 categories with 100 images each class, including diverse scene content such as fireworks, bark, microscopy images, tiles, trees, etc.	ACC [56], ACC Spatial Pyramid (ACC-Spy) [56], [70], Color and Edge Directivity Descriptor Spatial Pyramid (CEED-Spy) [69], [70], Convolutional Neural Network by Caffe [71] framework (CNN-Caffe), FCTH Spatial Pyramid (FCTH-Spy) [70], [78], Joint Composite Descriptor Spatial Pyramid (JCD-Spy) [67], [70], and Local Binary Patterns Spatial Pyramid (LBP-Spy) [70], [76]	MAP
UKBench [79]	10,200	Images: Objects/ Scenes	Composed of 2,550 objects or scenes. Each object/scene is captured 4 times from different viewpoints, distances, and illumination conditions.	ACC [56], BIC [57], Convolutional Neural Network by Caffe [71] framework (CNN-Caffe) Color and Edge Directivity Descriptor (CEED) [69], Fuzzy Color and Texture Histogram (FCTH) [78], FCTH Spatial Pyramid (FCTH-Spy) [70], [78], Joint Composite Descriptor (JCD) [67], Scale-Invariant Feature Transform (SIFT) [80], and Vocabulary Tree (VOC) [81]	N-S Score
ALOI [82]	72,000	Images: Objects	Images from 1,000 classes of objects, with different viewpoint and illumination.	ACC [56], BIC [57], GCH [58], Color Coherence Vectors (CCV) [83], and Local Color Histograms (LCH) [84]	MAP
MediaEval [85]	14,838	Videos	A total of 3,288 hours of video collected from blip.tv for the Video Genre Tagging Task at the MediaEval 2012. They are distributed among 26 genre categories.	Bag-of-Scenes (BoS) [86], and Histogram of Motion Patterns (HMP) [87], and Pooling over Pooling (PoP) [88]	MAP
FCVID [89]	87,648	Videos	A total of 4,232 hours of video collected from YouTube and annotated manually according to 233 categories.	Convolutional Neural Network (CNN) [90], Improved Dense Trajectories (IDT) [91], Mel-Frequency Cepstral Coefficients (MFCC) [92], Scale-Invariant Feature Transform (SIFT) [80]. Descriptors computed for each trajectory (IDT): Histogram of Oriented Gradients (IDT-HOG), Histogram of Optical Flow (IDT-HOF), Motion Boundary Histogram (IDT-MBH), and Trajectory Shape Descriptor (IDT-TRAJ).	MAP

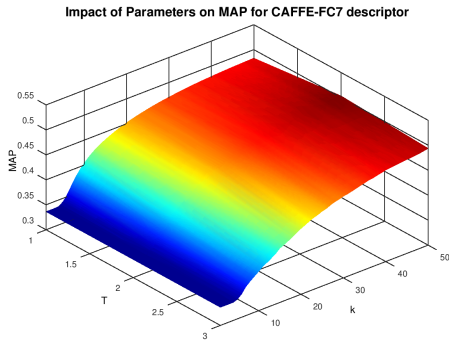


Fig. 4: Impact of variation of k and T on effectiveness - Corel5K [77] dataset.

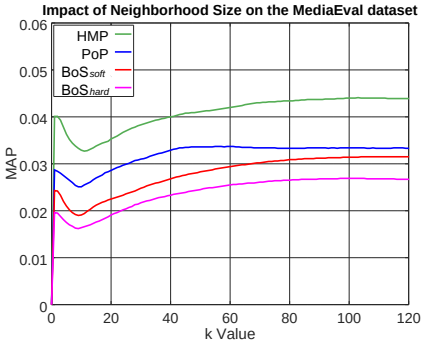


Fig. 5: Impact of neighborhood size (k) on effectiveness - MediaEval [85] dataset.

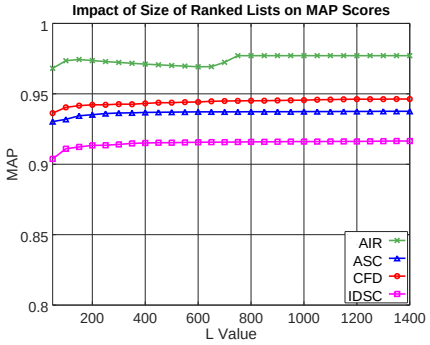


Fig. 6: Impact of ranked list size (L) on effectiveness - MPEG7 [59] dataset.

TABLE II: Effectiveness results on diverse image retrieval scenarios.

Dataset	Descriptor	Original MAP	LHRR (L=200) Fixed k	LHRR (L=200) Best k	k	LHRR (Full L) Fixed k	LHRR (Full L) Best k	k	Relative Gain
Soccer	GCH	32.24%	33.30%	33.55%	25	36.27%	**36.36%**	24	+12.78%
	ACC	37.23%	45.23%	47.40%	31	48.17%	**49.36%**	31	+32.58%
	BIC	39.26%	44.78%	48.02%	35	47.88%	**49.83%**	34	+26.92%
	BIC+ACC	-	46.41%	48.94%	35	49.03%	**50.35%**	34	-
MPEG-7	SS	37.67%	53.15%	53.40%	22	54.41%	**54.56%**	21	+44.84%
	BAS	71.52%	83.26%	83.30%	19	84.33%	**84.50%**	19	+18.15%
	CFD	80.71%	94.22%	94.22%	20	94.63%	**94.64%**	21	+17.26%
	IDSC	81.70%	91.34%	91.34%	20	91.65%	**91.69%**	21	+12.23%
	ASC	85.28%	93.52%	93.60%	18	93.76%	**93.81%**	18	+10.00%
	AIR	89.39%	97.36%	**97.77%**	28	97.71%	97.71%	20	+9.37%
	CFD+ASC	-	99.24%	**99.37%**	18	99.24%	**99.37%**	18	-
	CFD+AIR	-	99.92%	**99.97%**	19	99.92%	**99.97%**	19	-
	CFD+ASC+AIR	-	99.96%	**100%**	17	99.96%	**100%**	17	-
Brodatz	LBP	48.40%	49.84%	51.68%	10	50.49%	**52.57%**	10	+8.62%
	CCOM	57.57%	66.45%	66.63%	17	67.22%	**67.60%**	16	+17.42%
	LAS	75.15%	80.50%	81.63%	14	80.99%	**82.19%**	14	+9.37%
	CCOM+LAS	-	83.97%	84.36%	17	83.97%	**84.55%**	16	-

TABLE III: Effectiveness results on Corel5K [77] dataset.

Descriptor	Original MAP	LHRR (Full L)			Relative Gain
		Fixed k	Best k	k	
LBP-Spy	16.28%	19.02%	**20.60%**	52	+26.54%
ACC	27.75%	38.30%	**43.49%**	69	+56.72%
FCTH-Spy	27.89%	33.30%	**35.22%**	49	+26.28%
CNN-Caffe	28.07%	45.09%	**50.95%**	70	+81.51%
JCD-Spy	29.18%	35.48%	**38.23%**	46	+31.01%
ACC-Spy	29.76%	37.05%	**39.87%**	51	+33.97%
CEDD-Spy	30.01%	37.88%	**41.12%**	54	+37.02%
CEDD-Spy+ CNN	-	61.41%	**69.12%**	54	-
ACC+CNN	-	62.48%	**72.39%**	64	-
ACC+CNN+ CEDD-Spy	-	65.62%	**73.34%**	55	-

TABLE IV: Effectiveness results on Holidays [66] dataset.

Descriptor	Original MAP	LHRR (Full L)			Relative Gain
		Fixed k	Best k	k	
JCD-Spy	56.58%	**59.17%**	**59.17%**	5	+4.58%
FCTH-Spy	55.38%	**59.98%**	**59.98%**	5	+8.31%
CEDD-Spy	56.09%	58.73%	**59.24%**	4	+5.62%
ACC-Spy	62.37%	**67.21%**	**67.21%**	5	+7.76%
CNN-Caffe	64.09%	**70.81%**	**70.81%**	5	+10.49%
ACC	64.29%	**71.61%**	**71.61%**	5	+11.39%
$\text{VGG}_{dense,max}$	78.35%	**82.30%**	**82.30%**	5	+5.04%
CNN-Overfeat	82.59%	**85.54%**	**85.54%**	5	+3.57%
CNN-OLDFP	88.46%	**89.78%**	**89.78%**	5	+1.49%
CNN-OL+VGG	-	**89.00%**	**89.00%**	5	-
CNN-OL+CNN-Ov+VGG	-	**90.59%**	**90.59%**	5	-
CNN-OL+CNN-Ov	-	**90.94%**	**90.94%**	5	-

TABLE V: Effectiveness results on UKBench [79] dataset.

Descriptor	Original Score	LHRR (L=200)			Relative Gain
		Fixed k	Best k	k	
SIFT	2.54	3.10	**3.11**	6	+22.44%
CEDD	2.61	2.81	**2.82**	6	+8.05%
FCTH	2.73	2.88	**2.90**	6	+6.23%
JCD	2.79	2.99	**3.00**	6	+7.53%
BIC	3.04	3.27	**3.28**	6	+7.89%
HSV3D	3.17	**3.40**	**3.40**	5	+7.26%
CNN-Caffe	3.31	3.63	**3.63**	6	+9.67%
COMO	3.33	**3.55**	**3.55**	5	+6.61%
ACC	3.36	**3.65**	**3.65**	5	+8.63%
VOC	3.54	3.78	**3.78**	6	+6.78%
$\text{VGG}_{dense,max}$	3.65	**3.86**	**3.86**	5	+5.75%
CNN-OLDFP	3.84	**3.94**	**3.94**	5	+2.60%
ACC+VOC+CNN-Caffe	-	**3.93**	**3.93**	5	-
CNN-OL+VGG	-	**3.94**	**3.94**	5	-
CNN-OL+VGG+VOC	-	**3.96**	**3.96**	5	-

which has an original MAP of 28.07% is improved to 50.95% by the LHRR method. While the best isolated descriptor achieves a MAP of 30.01%, the best fusion computed by LHRR achieves 75.34%.

The UKBench [79] and the Holidays [66] dataset are very challenging due to the small number of images per class. Table IV presents the results for Holidays [66] dataset considering MAP score, while Table V presents the effectiveness results for the UKBench [79] considering the N-S score. The N-S score is computed between 1 and 4, which corresponds to the number of relevant images among the first four image returned (or P@4). Despite of the challenging scenario, positive gains can be observed for both datasets, reaching +22.44% and very high-effective scores of 90.94% and 3.96. Considering two iterations, the LHRR achieves 3.97 for the best fusion combination on UKBench [79] dataset.

E. Object Retrieval

The LHRR method is evaluated in object retrieval tasks considering the ALOI [82] dataset. Table VI presents the

TABLE VI: Effectiveness results for the ALOI [82] dataset.

Descriptor	Original MAP	LHRR (L=3000)			Relative Gain
		Fixed k	Best k	k	
ACC	43.77%	**55.06%**	**55.06%**	20	+25.79%
CCV	47.49%	55.32%	**55.97%**	29	+17.86%
GCH	50.56%	60.94%	**61.05%**	22	+20.75%
LCH	58.55%	83.80%	**83.83%**	21	+43.18%
BIC	71.75%	86.19%	**87.51%**	31	+21.97%
LCH+BIC	-	86.48%	**88.42%**	34	-

TABLE VII: Effectiveness results on MediaEval [85] dataset.

Descriptor	Original MAP	HyperGraph (L=1000)			Relative Gain
		Fixed k	Best k	k	
BoS_{hard}	1.76%	2.61%	**2.69%**	97	+52.84%
BoS_{soft}	2.23%	3.02%	**3.15%**	104	+41.26%
PoP	2.53%	3.34%	**3.37%**	55	+33.20%
HMP	3.85%	4.30%	**4.41%**	103	+14.55%
HMP+PoP	-	5.19%	**5.21%**	82	-

TABLE VIII: Effectiveness results on FCVID [89] dataset.

Descriptor	Original MAP	LHRR (L=1000)			Relative Gain
		Fixed k	Best k	k	
MFCC	1.77%	2.37%	**2.5%**	33	+41.24%
SIFT	2.24%	**3.74%**	**3.74%**	70	+66.96%
IDT-TRAJ	2.73%	3.46%	**3.46%**	68	+26.74%
IDT-HOF	3.65%	5.33%	**5.33%**	69	+46.03%
IDT-HOG	3.80%	6.35%	**6.35%**	66	+67.11%
IDT-MBH	4.61%	**7.48%**	**7.48%**	70	+62.26%
CNN	8.42%	**17.65%**	**17.65%**	70	+109.62%
CNN+IDT-HOG	-	6.06%	**6.06%**	70	-

effectiveness results considering the MAP scores for two different values of L. Very significant effectiveness gains can be observed for all descriptors. For instance, the LCH descriptor, which presented an initial score of 58.55%, achieved 83.83% using the LHRR method.

F. Video Retrieval

The LHRR method was also evaluated in video retrieval tasks considering two different datasets. Tables VII and VIII present the effectiveness results, respectively for the MediaEval [85] and FCVID [89] datasets. Again, remarkable gains are obtained for various features, specially for CNN on FCVID [89], which was improved by the LHRR methdo from 8.42% to 17.65%.

G. Discussion and Analysis

In addition to effectiveness results, other aspects of the proposed method are also analyzed. Firstly, we investigated convergence aspects, measuring the rank correlation between iterations and the evolution of effectiveness measures. Figure 7 illustrates the results, considering the two features of the MPEG-7 dataset. The Kendall τ measure is considered as rank correlation measure (high values indicate similarity) and the MAP as effectiveness measure. We can observe that the highest effectiveness gain is obtained at the first iteration and the rankings converges to a stable state, as indicates the Kendall τ values.

A complementary view to such analysis is illustrated in Figure 8. The evolution of average hyperedge weights along iterations is illustrated in conjunction with a effectiveness measure (P@20). As expected, the hyperedge weights, which estimate the effectiveness of ranked lists, grows along iterations similarly to the P@20 measure.

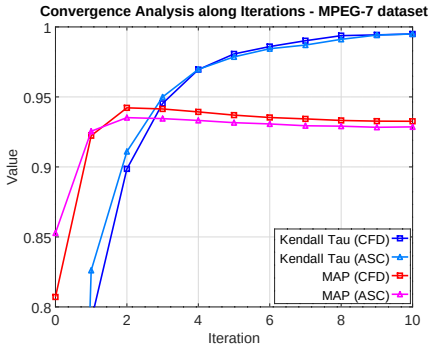


Fig. 7: Convergence analysis by measuring rank correlation between iterations.

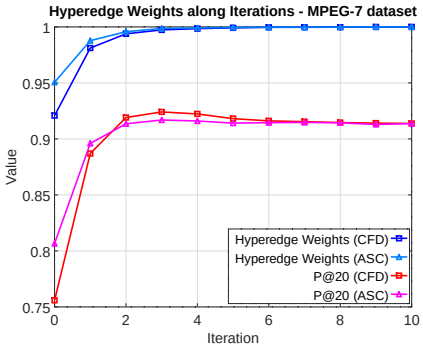


Fig. 8: Hyperedge weights evolution according to iterations.

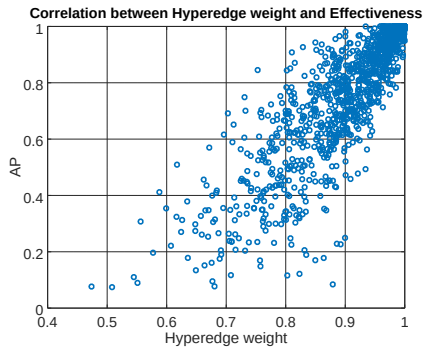


Fig. 9: Correlation between hyperedge weight and effectiveness measure (AP).

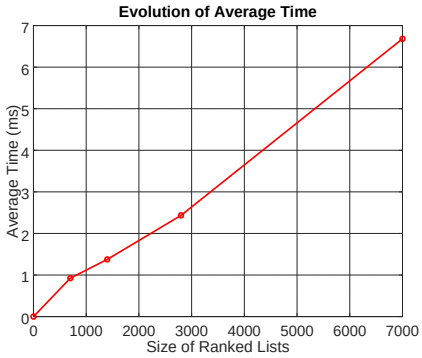


Fig. 10: Scalability analysis: average time per query on the ALOI [82] dataset.

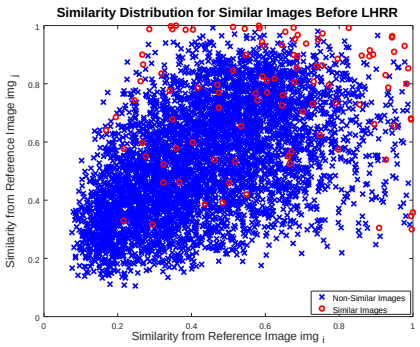


Fig. 11: Similarity distribution before the LHRR algorithm execution.

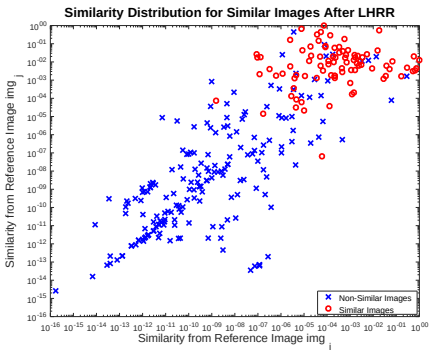


Fig. 12: Similarity distribution after the LHRR algorithm execution.

TABLE IX: Individual impact of LHRR steps on the MAP.

Dataset	Descriptor	Hyperedges Similarities	Cartesian Product	Full Allgorithm
Soccer	BIC	+21.12%	+10.56%	**+21.60%**
MPEG-7	AIR	+8.32%	+7.80%	**+9.31%**
Holidays	CNN-ODLFP	**+1.75%**	+0.54%	+1.49%
Brodatz	LAS	+7.55%	+4.72%	**+7.66%**
Corel5k	CNN-DF	+56.93%	+42.00%	**+60.63%**
UKBench	CNN-ODLFP	**+1.28%**	+1.22%	+1.15%
ALOI	BIC	+19.01%	**+20.61%**	+20.42%
Mean	-	+16.57%	+12.49%	**+17.47%**

The correlation between the hyperedge weights and the effectiveness measure can also be observed in Figure 9. Each image is illustrated by a point in the graph, where the x axis refers to the hyperedge weight and the y axis refers to the MAP measure. A high correlation can be observed, demonstrating that the hyperedge weight is an effective unsupervised estimation of effectiveness.

The impact of each step of the algorithm is also evaluated. We evaluate the effectiveness of hyperedges similarities and the Cartesian product in isolation compared with the full algorithm. The most effective feature of each dataset is considered in the experiment. Table IX presents the results. As we can observe, the full algorithm achieved the best effectiveness results in most of situations.

Finally, efficiency aspects are also analyzed. An experiment was conducted in ALOI [82] dataset, varying the size of ranked lists and measuring the execution time. Figure 10 shows the average time per query according to different ranked lists size. Notice that a linear behavior can be observed, demonstrating the scalability of the method.

H. Visual Results

In addition to the extensive quantitative evaluation, a visual analysis is also presented in this section. First, we evaluate the impact of the proposed LHRR method on the similarity distribution, considering the Corel5K dataset. A bidimensional representation of a dataset before and after the execution of the algorithm is considered. The representation is constructed based on two images, arbitrarily selected and called as reference images. Next, all collection images are represented in the bidimensional space, such that their position is defined according to their distance to the reference images.

The bidimiensional representation illustrating the similarity distribution of Corel5K [77] dataset before the algorithm execution is shown in Figure 11. The representation considers the score obtained after the rank normalization procedure. Similar images to the reference images are illustrated in red circles and remaining images in blue. As we can observe, similar and non-similar images are mixed in the similarity space, leading to low-effective retrieval results. Figure 12 illustrates the similarity distribution after the use of the LHRR method. The capacity of taking into account the dataset manifold in order to increase the separability between similar and non-similar images is remarkable.

The impact of the new similarity distributions on the effectiveness of ranking tasks is illustrated in Figure 13. The reference images (img_i and img_j) are illustrated with green borders at left and taken as queries. The figure illustrates the ranked lists obtained before and after the use of the LHRR,

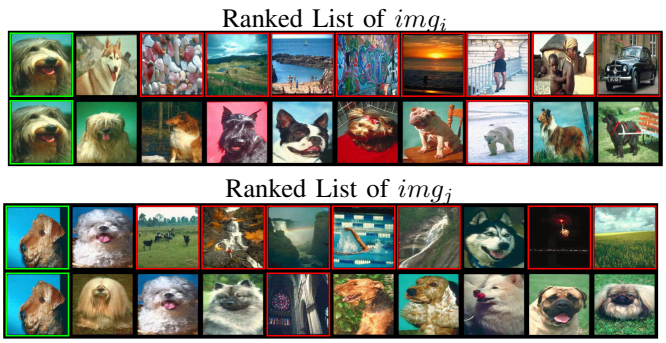


Fig. 13: Visual examples from the Corel5K dataset.

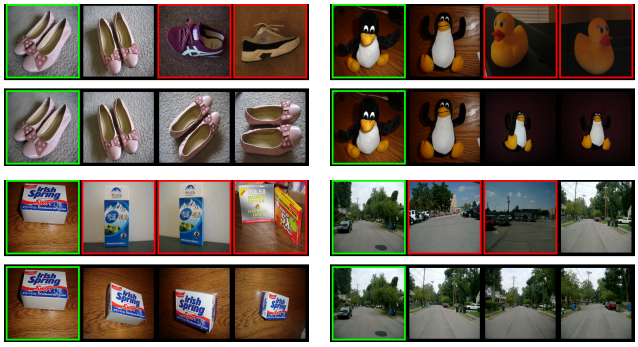


Fig. 14: Visual examples from the UKBench [79] dataset.

evincing the impressive gains in effectiveness. Other visual examples of UKBench [79] dataset are illustrated in Figure 14.

I. Comparison with Other Approaches

The LHRR method is also evaluated in comparison with various state-of-the-art post-processing methods and retrieval approaches. Different aspects are considered, with experiments conducted on three datasets: MPEG-7 [59], Holidays [66] and UKBench [79], which are popular datasets commonly used as benchmark for image retrieval and post-processing methods.

Table X presents the results on the MPEG-7 [59] dataset, considerig the bull's eye score (Recall@40), which counts all matching shapes within the top-40 ranked images, as evaluation measure. Four different features (IDSC, CFD, ASC, and AIR) and several recent post-processing methods are evaluated. Notice that the LHRR achieved high-effective scores for all features, reaching the best result for three of them.

The comparison of effectiveness results on the Holidays [66] dataset is presented in Table XI. Various state-of-the-art retrieval approaches are included in the comparison and the LHRR achieves a high-effective result of **90.94%**. Once diverse features are used by the related approaches, a comparison considering the same features used by the LHRR method is included as baseline. The Graph Fusion [20], which is a relevant unsupervised method is considered. The neighborhood size is used as $k = 5$ and fusion tasks was perfomed by the graph density variation [20].

Table XII presents the comparison on the UKBench [79] dataset, considering recent state-of-the-art retrieval approaches. The results of LHRR are reported for various features, also including the Graph Fusion [20] method as baseline. The proposed LHRR method yielded a very high-effective N-S score of **3.96**.

TABLE X: Comparison with various post-processing methods on the MPEG-7 [59] dataset (Bull's eye score - Recall@40).

Shape Descriptors		**Bull's eye score**
Contour Feat. Descriptor (CFD) [63]	-	84.43%
Inner Dist. Shape Context (IDSC) [64]		85.40%
Aspect Shape Context (ASC) [61]		88.39%
Articulation-Invariant Rep. (AIR) [60]		93.67%
Post-Processing Methods	**Descriptors**	**Bull's eye score**
Contextual Dissimilarity Measure [51]	IDSC [64]	88.30%
Graph Transduction [93]		91.00%
Self-Smoothing Operator [5]		92.77%
Local Constr. Diff. Process [17]		93.32%
Mutual kNN Graph [94]		93.40%
SCA [9]		93.44%
Smooth Neighborhood [95]		93.52%
Reciprocal kNN Graph CCs [10]		93.62%
Proposed LHRR		**94.21%**
Graph Fusion [20]	CFD [63]	89.76%
Index-Based Re-Ranking [53]		92.85%
RL-Sim [96]		94.27%
Correlation Graph [24]		94.84%
Reciprocal kNN Graph CCs [10]		96.51%
Proposed LHRR		**97.02%**
Generic Diffusion Process [16]	ASC [61]	93.95%
Index-Based Re-Ranking [53]		94.09%
Correlation Graph [24]		95.50%
Local Constr. Diff. Process [17]		95.96%
Smooth Neighborhood [95]		95.98%
Reciprocal kNN Graph CCs [10]		96.04%
Proposed LHRR		**96.36%**
Tensor Product Graph [97]		96.47%
Graph Fusion [20]	AIR [60]	98.76%
Index-Based Re-Ranking [53]		99.93%
RL-Sim [96]		99.94%
Tensor Product Graph [97]		99.99%
Generic Diffusion Process [16]		100%
Neighbor Set Similarity [18]		100%
Reciprocal kNN Graph CCs [10]		100%
Proposed LHRR		**100%**

TABLE XI: Comparison with state-of-the-art on the Holidays [66] dataset (MAP score).

MAP scores for state-of-the-art methods.				
Tolias *et al.* [98]	Paulin *et al.* [99]	Qin *et al.* [100]	Zheng *et al.* [101]	Sun *et al.* [3]
82.20%	82.90%	84.40%	85.20%	85.50%

Zheng *et al.* [102]	Pedronette et al. [10]	Iscen et al. [21]	Li *et al.* [103]	Liu et al. [7]
85.80%	86.19%	87.5%	89.20%	90.89 %

MAP scores for the proposed method		
Descriptor	**Baseline:** Graph Fusion [20]	**Proposed:** LHRR
ACC	66.42%	71.61%
CNN-Caffe	66.79%	70.81%
CNN-Overfeat	83.79%	85.54%
CNN-OLDFP	89.00 %	89.15%
ACC+CNN-Caffe	71.02%	81.84%
ACC+CNN-Overfeat	76.55%	86.35%
ACC+CNN-Caffe+CNN-Overfeat	80.06%	87.62%
CNN-OLDFP+CNN-Overfeat	79.36%	**90.94%**

TABLE XII: Comparison with state-of-the-art on the UK-Bench [79] dataset (N-S score - P@4).

N-S scores for state-of-the-art methods				
Wang et al. [104]	Sun et al. [3]	Paulin *et al.* [99]	Zhang et al. [20]	Zheng et al. [52]
3.68	3.76	3.76	3.83	3.84
Bai et al. [9]	Xie et al. [105]	Liu et al. [7]	Pedronette et al. [10]	Bai et al. [15]
3.86	3.89	3.92	3.93	3.93

N-S scores for the proposed method		
Descriptor	**Baseline:** Graph Fusion [20]	**Proposed:** LHRR
ACC	3.48	3.65
CNN-Caffe	3.45	3.63
VOC	3.67	3.78
CNN-OLDFP	3.87	3.94
ACC+CNN-Caffe	3.70	3.86
ACC+VOC	3.78	3.87
VOC+CNN-Caffe	3.78	3.89
ACC+VOC+CNN-Caffe	3.86	3.93
CNN-OLDFP+VGG+VOC	3.90	**3.96**

VII. Conclusions

Accurately ranking have been established as a task of paramount importance for retrieval applications. In this paper, a novel manifold ranking algorithm was proposed based on hypergraphs. The LHRR algorithm exploits the capacity of hypergrahs of modelling high-order similarity relationships to analyze the dataset manifold A contextual representation is proposed based on hyperedges and is used to compute more effective retrieval results.

The method was extensively evaluated considering diverse multimedia retrieval scenarios. High effective results were achieved in comparison with state-of-the art. As future work, we intend to investigate the LHRR method for multimodal retrieval and other learning tasks, involving supervised and semi-supervised scenarios.

Acknowledgments

We thank the São Paulo Research Foundation - FAPESP (grants #2018/15597-6, #2017/25908-6, #2017/02091-4, #2017/20945-0, #2016/06441-7, #2015/24494-8, #2016/50250-1, #2013/50155-0, #2014/12236-1, and #2014/50715-9), the Brazilian National Council for Scientific and Technological Development - CNPq (grants #423228/2016-1, #307560/2016-3, #308194/2017-9, and #313122/2017-2), and the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

References

[1] W. Zhou, H. Li, and Q. Tian, "Recent advance in content-based image retrieval: A literature survey," *CoRR*, vol. abs/1706.06064, 2017.

[2] F. Radenovic, A. Iscen, G. Tolias, Y. S. Avrithis, and O. Chum, "Revisiting Oxford and Paris: Large-scale image retrieval benchmarking," in *IEEE Int. Conf. Computer Vision and Pattern Recognition (CVPR2018)*, 2018.

[3] S. Sun, Y. Li, W. Zhou, Q. Tian, and H. Li, "Local residual similarity for image re-ranking," *Information Sciences*, vol. 417, no. Sup. C, pp. 143 – 153, 2017.

[4] Y. Wang, W. Zhang, L. Wu, X. Lin, and X. Zhao, "Unsupervised metric fusion over multiview data by graph random walk-based cross-view diffusion," *IEEE Trans. on Neural Nets. and Learning Systems*, vol. 28, no. 1, pp. 57–70, 2017.

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

[6] W. Li, Y. Wu, and J. Li, "Re-identification by neighborhood structure metric learning," *Pattern Recognition*, vol. 61, pp. 327 – 338, 2017.

[7] Z. Liu, S. Wang, L. Zheng, and Q. Tian, "Robust imagegraph: Rank-level feature fusion for image search," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3128–3141, 2017.

[8] J. Garca, N. Martinel, A. Gardel, I. Bravo, G. L. Foresti, and C. Micheloni, "Discriminant context information analysis for post-ranking person re-identification," *IEEE Transactions on Image Processing*, vol. 26, no. 4, pp. 1650–1665, 2017.

[9] S. Bai and X. Bai, "Sparse contextual activation for efficient visual re-ranking," *IEEE Trans. on Image Processing (TIP)*, vol. 25, no. 3, pp. 1056–1069, 2016.

[10] D. C. G. Pedronette, F. M. F. Gonalves, and I. R. Guilherme, "Unsupervised manifold learning through reciprocal kNN graph and Connected Components for image retrieval tasks," *Pattern Recognition*, vol. 75, pp. 161 – 174, 2018.

[11] A. Iscen, G. Tolias, Y. Avrithis, T. Furon, and O. Chum, "Efficient diffusion on region manifolds: Recovering small objects with compact cnn representations," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[12] L. Ma, H. Li, F. Meng, Q. Wu, and L. Xu, "Manifold-ranking embedded order preserving hashing for image semantic retrieval," *Journal of Visual Communication and Image Representation*, vol. 44, no. Sup C, pp. 29 – 39, 2017.

[13] Z. Zhong, L. Zheng, D. Cao, and S. Li, "Re-ranking person re-identification with k-reciprocal encoding," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'2017)*, 2017.

[14] S. Bai, Z. Zhou, J. Wang, X. Bai, L. J. Latecki, and Q. Tian, "Ensemble diffusion for retrieval," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 774–783.

[15] S. Bai, X. Bai, Q. Tian, and L. J. Latecki, "Regularized diffusion process on bidirectional context for object retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2018, on-line, to appear.

[16] M. Donoser and H. Bischof, "Diffusion processes for retrieval revisited," in *Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2013, pp. 1320–1327.

[17] 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*, 2009, pp. 357–364.

[18] 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.

[19] D. C. G. Pedronette, J. Almeida, and R. da S. Torres, "A graph-based ranked-list model for unsupervised distance learning on shape retrieval," *Pattern Recognition Letters*, vol. 83, no. Part 3, pp. 357 – 367, 2016.

[20] S. Zhang, M. Yang, T. Cour, K. Yu, and D. Metaxas, "Query specific rank fusion for image retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 4, pp. 803–815, April 2015.

[21] A. Iscen, G. Tolias, Y. S. Avrithis, and O. Chum, "Mining on Manifolds: Metric learning without labels," in *IEEE Int. Conf. Computer Vision and Pattern Recognition (CVPR2018)*, 2018.

[22] A. Iscen, Y. S. Avrithis, G. Tolias, T. Furon, and O. Chum, "Fast spectral ranking for similarity search," in *IEEE Int. Conf. Computer Vision and Pattern Recognition (CVPR2018)*, 2018.

[23] J. Xu, C. Wang, C. Qi, C. Shi, and B. Xiao, "Iterative manifold embedding layer learned by incomplete data for large-scale image retrieval," *CoRR*, vol. abs/1707.09862, 2017.

[24] D. C. G. Pedronette and R. da Silva Torres, "A correlation graph approach for unsupervised manifold learning in image retrieval tasks," *Neurocomputing*, vol. 208, pp. 66–79, 2016.

[25] P. Jing, Y. Su, C. Xu, and L. Zhang, "Hyperssr: A hypergraph based semi-supervised ranking method for visual search reranking," *Neurocomputing*, 2016.

[26] B. Wei, M. Cheng, C. Wang, and J. Li, "Combinative hypergraph learning for semi-supervised image classification," *Neurocomputing*, vol. 153, no. Sup. C, pp. 271 – 277, 2015.

[27] Y. Huang, Q. Liu, S. Zhang, and D. N. Metaxas, "Image retrieval via probabilistic hypergraph ranking," in *IEEE Conference on Conference on Computer Vision and Pattern Recognition (CVPR'10)*, June 2010, pp. 3376–3383.

[28] Q. Liu, Y. Huang, and D. N. Metaxas, "Hypergraph with sampling for image retrieval," *Pattern Recognition*, vol. 44, no. 10, pp. 2255 – 2262, 2011, semi-Supervised Learning for Visual Content Analysis and Understanding.

[29] L. Zheng, Y. Yang, and Q. Tian, "Sift meets cnn: A decade survey of instance retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 5, pp. 1224 – 1244, 2017.

[30] Q. Liu, Y. Sun, C. Wang, T. Liu, and D. Tao, "Elastic net hypergraph learning for image clustering and semi-supervised classification," *IEEE Transactions on Image Processing*, vol. 26, no. 1, pp. 452–463, Jan 2017.

[31] J. Cai, Z. J. Zha, M. Wang, S. Zhang, and Q. Tian, "An attribute-assisted reranking model for web image search," *IEEE Transactions on Image Processing*, vol. 24, no. 1, pp. 261–272, Jan 2015.

[32] X. Cheng, Y. Zhu, J. Song, G. Wen, and W. He, "A novel low-rank hypergraph feature selection for multi-view classification," *Neurocomputing*, vol. 253, no. Supp. C, pp. 115 – 121, 2017.

[33] J. Yu, D. Tao, and M. Wang, "Adaptive hypergraph learning and its application in image classification," *IEEE Transactions on Image Processing*, vol. 21, no. 7, pp. 3262–3272, July 2012.

[34] L. Zhu, J. Shen, L. Xie, and Z. Cheng, "Unsupervised topic hypergraph hashing for efficient mobile image retrieval," *IEEE Transactions on Cybernetics*, vol. 47, no. 11, pp. 3941–3954, 2017.

[35] Y. Gao, M. Wang, D. Tao, R. Ji, and Q. Dai, "3-d object retrieval and recognition with hypergraph analysis," *IEEE Transactions on Image Processing*, vol. 21, no. 9, pp. 4290–4303, 2012.

[36] Y. Huang, *Graph-Based Methods in Computer Vision: Developments and Applications*. IGI Global, 2013, ch. Hypergraph Based Visual Segmentation and Retrieval, pp. 118–139.

[37] A. Bretto, *Hypergraph Theory: An Introduction*. Springer Publish. Company, 2013.

[38] B. Schlkopf, J. Platt, and T. Hofmann, "Learning with hypergraphs: Clustering, classification, and embedding," in *Advances in Neural Information Processing Systems (NIPS'07)*, 2007, pp. 1601–1608.

[39] L. Sun, S. Ji, and J. Ye, "Hypergraph spectral learning for multi-label classification," in *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '08, 2008, pp. 668–676.

[40] S. Agarwal, J. Lim, L. Zelnik-Manor, P. Perona, D. Kriegman, and S. Belongie, "Beyond pairwise clustering," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 2, June 2005, pp. 838–845 vol. 2.

[41] W. Wang, S. Li, J. Li, W. Li, and F. Wei, "Exploring hypergraph-based semi-supervised ranking for query-oriented summarization," *Information Sciences*, vol. 237, no. Sup. C, pp. 271 – 286, 2013.

[42] L. An, X. Chen, and S. Yang, "Person re-identification via hypergraph-based matching," *Neurocomputing*, vol. 182, no. Sup. C, pp. 247 – 254, 2016.

[43] Y. Gao, M. Wang, Z. J. Zha, J. Shen, X. Li, and X. Wu, "Visual-textual joint relevance learning for tag-based social image search," *IEEE Transactions on Image Processing*, vol. 22, no. 1, pp. 363–376, Jan 2013.

[44] X. Chen and J. Xu, "Uncooperative gait recognition: Re-ranking based on sparse coding and multi-view hypergraph learning," *Pattern Recognition*, vol. 53, no. Sup. C, pp. 116 – 129, 2016.

[45] W. He, X. Cheng, R. Hu, Y. Zhu, and G. Wen, "Feature self-representation based hypergraph unsupervised feature selection via low-rank representation," *Neurocomputing*, vol. 253, no. Sup. C, pp. 127 – 134, 2017.

[46] S. Huang, A. Elgammal, and D. Yang, "On the effect of hyperedge weights on hypergraph learning," *Image and Vision Computing*, vol. 57, no. Sup. C, pp. 89 – 101, 2017.

[47] J. Lu, G. Wang, W. Deng, P. Moulin, and J. Zhou, "Multi-manifold deep metric learning for image set classification," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1137–1145.

[48] X. Lv and F. Duan, "Metric learning via feature weighting for scalable image retrieval," *Pattern Recognition Letters*, 2017.

[49] H. Jia, Y. m. Cheung, and J. Liu, "A new distance metric for unsupervised learning of categorical data," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 5, pp. 1065–1079, 2016.

[50] I. Theodorakopoulos, G. Economou, S. Fotopoulos, and C. Theoharatos, "Local manifold distance based on neighborhood graph reordering," *Pattern Recognition*, vol. 53, pp. 195 – 211, 2016.

[51] H. Jegou, C. Schmid, H. Harzallah, and J. Verbeek, "Accurate image search using the contextual dissimilarity measure," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 1, pp. 2–11, 2010.

[52] 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 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[53] D. C. G. Pedronette, J. Almeida, and R. da S. Torres, "A scalable re-ranking method for content-based image retrieval," *Information Sciences*, vol. 265, no. 1, pp. 91–104, 2014.

[54] L. P. Valem and D. C. G. a. Pedronette, "An unsupervised distance learning framework for multimedia retrieval," in *ACM on International Conference on Multimedia Retrieval*, ser. ICMR '17, 2017, pp. 107–111.

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

[56] J. Huang, S. R. Kumar, M. Mitra, W.-J. Zhu, and R. Zabih, "Image indexing using color correlograms," in *CVPR'97*, 1997, pp. 762–768.

[57] 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*, 2002, pp. 102–109.

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

[59] L. J. Latecki, R. Lakmper, and U. Eckhardt, "Shape descriptors for non-rigid shapes with a single closed contour."

[60] R. Gopalan, P. Turaga, and R. Chellappa, "Articulation-invariant representation of non-planar shapes," in *11th European Conference on Computer Vision (ECCV'2010)*, vol. 3, 2010, pp. 286–299.

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

[62] 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.

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

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

[65] 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.

[66] H. Jegou, M. Douze, and C. Schmid, "Hamming embedding and weak geometric consistency for large scale image search," in *European Conference on Computer Vision*, ser. ECCV '08, 2008, pp. 304–317.

[67] K. Zagoris, S. Chatzichristofis, N. Papamarkos, and Y. Boutalis, "Automatic image annotation and retrieval using the joint composite descriptor," in *14th Panhellenic Conference on Informatics (PCI)*, 2010, pp. 143–147.

[68] 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.

[69] S. A. Chatzichristofis and Y. S. Boutalis, "Cedd: color and edge directivity descriptor: a compact descriptor for image indexing and retrieval," in *Proceedings of the 6th international conference on Computer vision systems*, ser. ICVS'08, 2008, pp. 312–322.

[70] M. Lux, "Content based image retrieval with LIRe," in *Proceedings of the 19th ACM International Conference on Multimedia*, ser. MM '11, 2011.

[71] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. B. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," *CoRR - http://arxiv.org/abs/1408.5093*.

[72] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "CNN features off-the-shelf: an astounding baseline for recognition," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW'14)*, pp. 512–519.

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

[74] V. Kovalev and S. Volmer, "Color co-occurence descriptors for querying-by-example," in *International Conference on Multimedia Modeling*, 1998, p. 32.

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

[76] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971–987, 2002.

[77] "Content-based image retrieval using color difference histogram," *Pattern Recognition*, vol. 46, no. 1, pp. 188 – 198, 2013.

[78] S. A. Chatzichristofis and Y. S. Boutalis, "FCTH: Fuzzy color and texture histogram a low level feature for accurate image retrieval," in *Int. Workshop on Image Analysis for Multimedia Interactive Services*, 2008, pp. 191–196.

[79] D. Nistér and H. Stewénius, "Scalable recognition with a vocabulary tree," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'2006)*, vol. 2, 2006, pp. 2161–2168.

[80] D. Lowe, "Object recognition from local scale-invariant features," in *IEEE International Conference on Computer Vision (ICCV)*, 1999, pp. 1150–1157.

[81] X. Wang, M. Yang, T. Cour, S. Zhu, K. Yu, and T. Han, "Contextual weighting for vocabulary tree based image retrieval," in *IEEE International Conference on Computer Vision (ICCV'2011)*, Nov 2011, pp. 209–216.

[82] J.-M. Geusebroek, G. J. Burghouts, and A. W. M. Smeulders, "The amsterdam library of object images," *International Journal of Computer Vision*, vol. 61, no. 1, pp. 103–112, 2005.

[83] G. Pass, R. Zabih, and J. Miller, "Comparing images using color coherence vectors," in *ACM International Conference on Multimedia*, 1996, pp. 65–73.

[84] H. Lu, B. Ooi, and K. Tan, "Efficient image retrieval by color contents," in *Int. Conference on Applications of Databases (ADB)*, 1994, pp. 95–108.

[85] S. Schmiedeke, C. Kofler, and I. Ferrané, "Overview of mediaeval 2012 genre tagging task," in *MediaEval*, 2012.

[86] O. A. B. Penatti, L. T. Li, J. Almeida, and R. S. Torres, "A visual approach for video geocoding using bag-of-scenes," in *ACM International Conference on Multimedia Retrieval (ICMR)*, 2012, pp. 1–8.

[87] J. Almeida, N. J. Leite, and R. S. Torres, "Comparison of video sequences with histograms of motion patterns," in *IEEE International Conference on Image Processing (ICIP)*, 2011, pp. 3673–3676.

[88] J. Almeida, D. C. G. Pedronette, and O. A. B. Penatti, "Unsupervised manifold learning for video genre retrieval," in *Iberoamerican Congress on Pattern Recognition (CIARP)*, 2014, pp. 604–612.

[89] Y.-G. Jiang, Z. Wu, J. Wang, X. Xue, and S.-F. Chang, "Exploiting feature and class relationships in video categorization with regularized deep neural networks," *arXiv preprint arXiv:1502.07209*, 2015.

[90] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Annual Conference on Neural Information Processing Systems (NIPS)*, 2012, pp. 1106–1114.

[91] H. Wang and C. Schmid, "Action recognition with improved trajectories," in *IEEE International Conference on Computer Vision (ICCV)*, 2013, pp. 3551–3558.

[92] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. 28, no. 4, pp. 357–366, 1980.

[93] X. Yang, X. Bai, L. J. Latecki, and Z. Tu, "Improving shape retrieval by learning graph transduction," in *European Conference on Computer Vision (ECCV'2008)*, vol. 4, 2008, pp. 788–801.

[94] P. Kontschieder, M. Donoser, and H. Bischof, "Beyond pairwise shape similarity analysis," in *Asian Conf. on Computer Vision (ACCV'2009)*, 2009, pp. 655–666.

[95] S. Bai, S. Sun, X. Bai, Z. Zhang, and Q. Tian, "Smooth neighborhood structure mining on multiple affinity graphs with applications to context-sensitive similarity," in *European Conference on Computer Vision (ECCV)*, 2016, pp. 592–608.

[96] 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.

[97] X. Yang, L. Prasad, and L. Latecki, "Affinity learning with diffusion on tensor product graph," *IEEE Transactions on Pattern Analysis and Machine Intelligence,*, vol. 35, no. 1, pp. 28–38, 2013.

[98] G. Tolias, Y. Avrithis, and H. Jgou, "To aggregate or not to aggregate: Selective match kernels for image search," in *IEEE International Conference on Computer Vision (ICCV'2013)*, Dec 2013, pp. 1401–1408.

[99] M. Paulin, J. Mairal, M. Douze, Z. Harchaoui, F. Perronnin, and C. Schmid, "Convolutional patch representations for image retrieval: An unsupervised approach," *Int. Journal of Computer Vision*, 2017.

[100] D. Qin, C. Wengert, and L. V. Gool, "Query adaptive similarity for large scale object retrieval," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'2013)*, June 2013, pp. 1610–1617.

[101] L. Zheng, S. Wang, and Q. Tian, "Coupled binary embedding for large-scale image retrieval," *IEEE Transactions on Image Processing (TIP)*, vol. 23, no. 8, pp. 3368–3380, 2014.

[102] L. Zheng, S. Wang, Z. Liu, and Q. Tian, "Packing and padding: Coupled multi-index for accurate image retrieval," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'2014)*, June 2014, pp. 1947–1954.

[103] X. Li, M. Larson, and A. Hanjalic, "Pairwise geometric matching for large-scale object retrieval," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'2015)*, June 2015, pp. 5153–5161.

[104] B. Wang, J. Jiang, WeiWang, Z.-H. Zhou, and Z. Tu, "Unsupervised metric fusion by cross diffusion," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'2012)*, 2012, pp. 3013 –3020.

[105] L. Xie, R. Hong, B. Zhang, and Q. Tian, "Image classification and retrieval are one," in *ACM Int. Conf. on Multimedia Retrieval (ICMR)*, 2015, pp. 3–10.