



Beyond diffusion process: Neighbor set similarity for fast re-ranking



Xiang Bai, Song Bai, Xinggang Wang*

School of Electronic Information and Communications, Huazhong Univ. of Science and Technology, N1 Hall, 1037 Luoyu Road, Hongshan, Wuhan 430074, PR China

ARTICLE INFO

Article history:

Received 6 January 2014

Revised 20 May 2015

Accepted 4 July 2015

Available online 11 July 2015

Keywords:

Shape retrieval

Image retrieval

Re-ranking

Contextual information

Diffusion process

ABSTRACT

Measuring the similarity between two instances reliably, shape or image, is a challenging problem in shape and image retrieval. In this paper, a simple yet effective method called Neighbor Set Similarity (NSS) is proposed, which is superior to both traditional pairwise similarity and diffusion process. NSS makes full use of contextual information to capture the geometry of the underlying manifold, and obtains a more precise measure than the original pairwise similarity. Moreover, based on NSS, we propose a powerful fusion process to utilize the complementarity of different descriptors to further enhance the retrieval performance. The experimental results on MPEG-7 shape dataset, N-S image dataset and ORL face dataset demonstrate the effectiveness of the proposed method. In addition, the time complexity of NSS is much lower than diffusion process, which suggests that NSS is more suitable for large scale image retrieval than diffusion process.

© 2015 Elsevier Inc. All rights reserved.

1. Introduction

Shape or image retrieval is a fundamental issue in computer vision with many applications. Given a query, all the instances in the database are sorted in an ascending (or descending) order based on their dissimilarity (similarity) to the query. Then, the ranking list of the query is initialized, where the most similar instance occupies the top position. Many researchers have focused on designing robust, informative and discriminative descriptors [3,11,15,18–20,24,28,33] in order to achieve better retrieval performance. However, these basic methods totally ignore the structure of the underlying data manifold, thus cannot generate satisfactory retrieval results.

In order to capture the geometry of the underlying manifold, many context-sensitive similarity measures [9,13,30,31,36–42,47] are proposed to improve the retrieval accuracy. Diffusion process, one of the most representative branches in context-based re-ranking, starts with constructing a weighted graph base on the graph theory [6], and uses the nodes to represent the visual instances. The edge connecting two nodes represents their pairwise similarity. Diffusion process conducts a random walk to spread the similarity through the graph, in which a transition matrix is used. The transition matrix usually interprets the similarity after normalization as the transition probability from one node to another. The computation of transition probability is usually relevant to the local distribution of the data manifold, which makes diffusion process robust to noise and outliers.

It seems that diffusion process is an indispensable tool for improving retrieval performance. However, it also has a disadvantage of computational expensive. Some iterative methods [12,36,37] require many computational steps to converge. These

* Corresponding author. Tel.: +86 27 87543236; fax: +86 27 87543236.

E-mail addresses: xbai@hust.edu.cn (X. Bai), songbai@hust.edu.cn (S. Bai), xgwang@hust.edu.cn, wxghust@gmail.com (X. Wang).

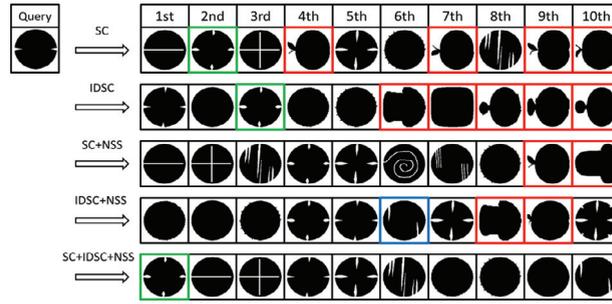


Fig. 1. Top to bottom: the retrieval results of a certain query from MPEG-7 dataset measured by SC, IDSC, NSS+SC, NSS+IDSC and NSS+IDSC+SC. The false retrieval results are put in red boxes. For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.

iterative methods are usually time-consuming, and improper for large scale re-ranking. Some other approaches [7,31] although have closed-form solutions, but they require complex operations, such as computing the inverse of matrices whose size is often proportional to the scale of the database. Such operations are computationally prohibited, when the size of database becomes larger. Such a shortcoming limits the usage of many diffusion-based algorithms in real-time search engines. Following the same principal as diffusion process, we propose Neighbor Set Similarity (NSS) to speed up the re-ranking procedure. Unlike the conventional approaches, NSS does not need an iterative process, resulting in much higher efficiency while keeping the re-ranking accuracy.

Besides the algorithms that focus on enhancing one type of similarity measure, some methods [1,30,32,44,45,48,49] are proposed to fuse multiple kinds of similarity measures for re-ranking, since different similarities may be complementary to each other. For example, as two popular shape descriptors, Shape Context (SC) [3] encodes the global information of a shape and generally works well with rigid objects, while Inner Distance Shape Context (IDSC) [15] replaces the Euclidean distance used in SC by the geodesic distance, and is more suitable for non-rigid analysis. It seems difficult to design a generic descriptor that can handle all the properties under different conditions, which inspires us to exploit a framework to fuse multiple complementary similarity measures. It is straightforward that a better performance can be achieved when the complementarity is used in a proper way. In this paper, based on NSS we propose a more powerful fusion method inspired by the co-training algorithm [5], which yields a much more precise retrieval result. However, unlike co-training that assumes views (sets of features) with two conditions, NSS deals with single-view but multiple-input similarity measures for robust re-ranking.

Fig. 1 shows the retrieval results when querying a given shape from MPEG-7 dataset [14], as measured by SC, IDSC, SC+NSS, IDSC+NSS and SC+IDSC+NSS. The false results are surrounded by red boxes. The first two rows show the retrieval results measured by SC and IDSC, and obviously several outliers exist in the ranking list. As the third row shows, NSS with SC as the input measure is more robust to noise compared with using SC only. The fourth result of SC is ranked in the 9th position in the retrieval result of NSS+SC, and some outliers are even excluded. Moreover, NSS is also able to find instances that are not in the original retrieval list. As the fourth row shows, the sixth result in the blue box is newly found, and it is not in the top-10 retrieval results of IDSC at first. What is more important is that NSS with two input measures can improve the retrieval results significantly by utilizing the complementarity of SC and IDSC. The instances that are ranked high in both measures will obtain a higher position in the retrieval result of NSS. For example, the instance in the green box, which holds the second position of SC and the third position of IDSC, is ranked first in the fifth row. This example shows NSS can utilize contextual information as well as multiple features to improve the retrieval performance.

The rest of this paper is organized as follows: In Section 2 we briefly revisit the related works. The motivation and definition of NSS are given in Section 3. A study of the comparison between NSS and diffusion process is given in Section 4. In Section 5, we conduct some experiments on several benchmark datasets to demonstrate the advantages of the proposed method again. Conclusions are given in Section 6.

2. Related work

In this section, we provide an overview of basic descriptors, diffusion process, feature fusion and kNN selection algorithms.

2.1. Descriptors

Many shape descriptors have been proposed in the literatures recently. Shape Context (SC) proposed in [3] works well for rigid objects, and Inner Distance Shape Context (IDSC) proposed in [15] is better at dealing with articulated shapes. Gopalan et al. [11] propose Articulation-Invariant Representation (AIR) by modeling an articulating shape as a combination of approximate convex parts connected by non-convex junctions. In [33], the contour of each shape is represented by a fixed number of sample points, and a height function is defined based on the distances of the other sample points to its tangent line. A more complicated matching method is introduced in [8], where SC is used to find the correspondence with dynamic programming,

then a modified version of edit distance is used to compute the similarity between strings of symbols that represent two contours. Gaussian mixture model is compared to find the best match given a query shape using Bregman divergence in [17].

However, the pairwise similarity is unable to capture category-level information across classes. Therefore, the similarity between two shapes can be correctly described only if it is considered in the context of other shapes similar to them. Thus, many context-based learning algorithms, especially diffusion processes introduced briefly in Section 2.2, are proposed to solve the problem.

2.2. Diffusion process

Diffusion process [9] firstly constructs an affinity matrix $W = [w_{ij}]_{n \times n}$, which relates N different database instances to each other. W is interpreted as a finite weighted graph $G = (V, E)$ consisting of N vertices $v_i \in V$ based on the dataset. The edge $e_{ij} \in E$ that links vertices is assigned to a nonnegative value W_{ij} . Most diffusion processes follow the same principle that they spread the affinity values through the graph, but define different initializations or transition matrices.

In [2,36], label propagation [10,50] is applied to shape/image retrieval, and Graph Transduction (GT) is proposed. GT considers the query itself as the only labeled data, and spreads the information from the labeled data to unlabeled data. In [37], Locally Constrained Diffusion Process (LCDP) defines that the diffusion process is restricted to the k -nearest neighbors (kNN) of the data points by replacing the full-connected graph G with a kNN-graph G_k . But if there are several noisy nodes, the paths through these nodes will affect the transition probability. In order to solve this problem, LCDP sets the transition probability to a high value if all the paths between the kNNs of the two vertices are short.

In [38], diffusion process is conducted in a graph obtained by the tensor product of the original graph with itself. Since Tensor Product Graph (TPG) takes into account the higher order information compared to the original graph, better retrieval performance can be obtained. But the higher order information requires for higher time and storage cost. Instead, the author proves that the propagation on TPG can be computed with the same computational complexity and the same amount of storage as the propagation on the original graph. Jegou et al. introduce Contextual Dissimilarity Measure (CDM) in [12] by taking the neighbors of an image into account. CDM is proper to improve the distance measures using Bag-of-Features (BoF) in image search, but does not work well with shape retrieval due to the fact that the property of shape distance measures is quite different from BoF used in image search.

Our proposed algorithm is similar to diffusion processes mentioned above, but it is more efficient and has the potential for large scale retrieval. NSS no longer needs the iterative procedure, and is more robust to noise in terms of statistics compared with diffusion process.

2.3. Feature fusion

Feature fusion is proven to be a strong tool for improving the performance due to the fact that different descriptors focus on different aspects of an object. In the specific scenario of retrieval, feature fusion can be applied at the indexing level or at the post-processing level. The representative algorithm of feature fusion at the indexing level is c-MI [45] that combines SIFT [20] and color feature. Each dimension of c-MI corresponds to one kind of feature. Multi-IDF is proposed in [46], in which different binary features are coupled into the inverted file. More researchers are dedicated to feature fusion at the post-processing level. The closest work to ours is Co-Transduction proposed in [1], which adopts a semi-supervised framework to fuse two complementary similarity measures. In [44], query specific fusion is proposed that fuses the ordered retrieval sets given by multiple retrieval methods. In [32], fusion process is exploited on a graph obtained by the tensor product of two different graphs.

The proposed NSS can be easily extended to similarity fusion, and also achieves better performance than diffusion-based similarity fusion algorithms as shown in Section 5.

2.4. kNN selection

The majority of the algorithms mentioned above require defining the context of an instance carefully. Some algorithms use the simplest definition: k -nearest neighbors (kNN), but kNN probably includes too much noise. Once the algorithm is not robust enough, the performance cannot be satisfactory. In order to obtain more accurate description for the context, a variant of kNN called Dominant Neighborhood (DN) is proposed in [38]. DN is a more robust version of kNN, which tries to maximize the average affinity between all pairs in kNN, and it does offer more faithful information compared with kNN, but the running time is largely increased, which is unbearable for large scale retrieval. In [25], a novel way to select the robust neighbors using the consensus of multiple rounds of kNNs is proposed. Consensus information can give better control over neighborhood selection. In the definition of NSS, our description for the context is the simple kNN, and NSS is indeed robust enough to decrease the negative effects brought by noise that exists in the kNN.

3. Neighbor set similarity

We first review the classical pipeline of shape or image retrieval concisely. Given a set of instances $\mathcal{X} = \{x_1, x_2, \dots, x_n\}$, the distance between x_q and x_p under a certain distance measure is defined $d(x_q, x_p)$, in which x_q represents the query instance, and x_p represents a certain database instance. For notation simplicity we will refer to x_q as q , and x_p as p for short in the whole paper.

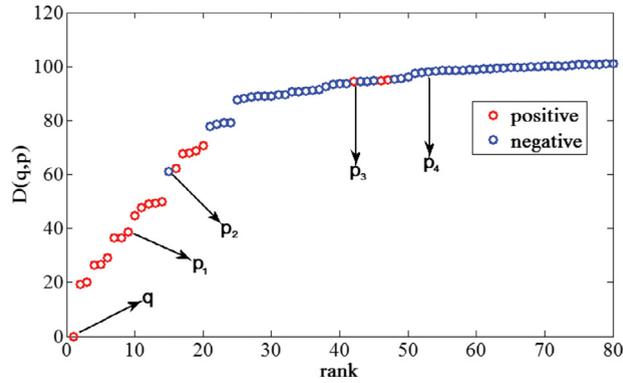


Fig. 2. The distribution of distance for a certain query q from the MPEG-7 dataset. The y-axis shows $d(q, p)$, while the x-axis shows the rank of returned shapes. The red circles represent true positives, p_1 (or p_3) is a delegate that a positive has a right (or false) ranking value, p_2 (or p_4) is a delegate that a negative has a false (or right) ranking value. For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.

By sorting the distance value $d(q, p)$ in ascending order for $p = 1, 2, \dots, n$, a ranking list is initialized based on the distance to the query q . Obviously, the most similar instance has the smallest value, and gets the top position in the ranking list.

In some cases, a similarity measure, which defines the similarity between q and p as $s(q, p)$, is used in place of the distance measure mentioned above. The retrieval is conducted differently, because the instance is sorted according to the similarity value to the query in decreasing order, and the most similar instance has the highest value.

As discussed above, the traditional measures cannot give a precise enough retrieval result. Our solution for improving the retrieval precision is to dig more faithful information from the original measure, not to design a perfect descriptor, which is also unrealistic.

3.1. Motivation

In order to show the motivation of our algorithm clearly, the set of k -nearest neighbors of q and p are denoted by $\mathcal{N}_k(q)$ and $\mathcal{N}_k(p)$. Our method is derived from a common sense that if the original $d(q, p)$ is small, but instances in $\mathcal{N}_k(q)$ are much different from instances in $\mathcal{N}_k(p)$, we can draw a conclusion that the occurrence of $d(q, p)$ is just incidental, or rather is false. On the contrary, if q and p are similar and also from the same class, we will find that most of their neighbors are also similar, although some noise and outliers may exist.

We take the retrieval results of MPEG-7 dataset measured by SC as an example. Each class in MPEG-7 dataset has 20 shapes, which means that a 100% retrieval precision is obtained, if all the other 19 shapes belonging to the same class as the query are in the top 20 retrieval list when querying a certain shape. In Fig. 2, we plot the distance value $d(q, p)$ for the top 80 ranked shapes for a given query q . The correctly retrieved instances are denoted by a red circle. The instances with extreme low values are of course true positives. But the curve becomes flat rapidly when the ranking value continues to increase, which gives the relevant instances and non-relevant instances almost the same distance values. It means the original distance measure only works well for instances close to the query q , which shows the necessity and importance of a learning method to further explore more faithful information based on the original measure.

According to the distance with a given query shape q , we divide the whole dataset roughly into four subsets: (1) $\mathcal{P}_1(q)$ represents the set of positive instances, and $d(q, p)$ ($p \in \mathcal{P}_1(q)$) is small enough. (2) $\mathcal{P}_2(q)$ is a set of negative instances with the value of $d(q, p)$ ($p \in \mathcal{P}_2(q)$) also small. The small values make these negative instances occupy positions in the top 20 retrieval list. (3) $\mathcal{P}_3(q)$ is defined as a set of positive instances, but $d(q, p)$ ($p \in \mathcal{P}_3(q)$) is relatively large, which gets these positive instances out of top-20 retrieval list. (4) The property of the set $\mathcal{P}_4(q)$ is that negative instances are far from the query q . An illustration of the classification is presented in Fig. 2, and p_i is a representative instance in $\mathcal{P}_i(q)$ ($1 \leq i \leq 4$).

It is obvious that we should pay attention to $\mathcal{P}_2(q)$ and $\mathcal{P}_3(q)$, for the instances in the two sets are both ranked in false positions. An excellent learning method should lower the ranking values of the instances in $\mathcal{P}_2(q)$, and lift those of instances in $\mathcal{P}_3(q)$. We continue to make use of the retrieval results that SC produces on the MPEG-7 dataset, and find that a learning method based on the contextual information is possible to solve the problem. SC gives us a top 20 precision 79.71%, and we define ε_{qp} as:

$$\varepsilon_{qp} = \frac{\frac{1}{k^2} \sum_{x_i \in \mathcal{N}_k(q)} \sum_{x_j \in \mathcal{N}_k(p)} d(x_i, x_j)}{d(q, p) \times (1 + \eta)} \quad (1)$$

to better reveal our motivation, where η is a slack variable whose value is rather small. ε_{qp} is a variable that can reveal the local distribute of q and p at the distance level.

Considering the instances in $\mathcal{P}_2(q)$, the distances between x_i and x_j ($i \in \mathcal{N}_k(q)$, $j \in \mathcal{N}_k(p)$) almost take a larger value compared with the original $d(q, p)$ ($p \in \mathcal{P}_2(q)$). Here we set η to -0.1 , -0.01 and 0 , and compute the percentage of $\varepsilon_{qp} \geq 1$ on the whole

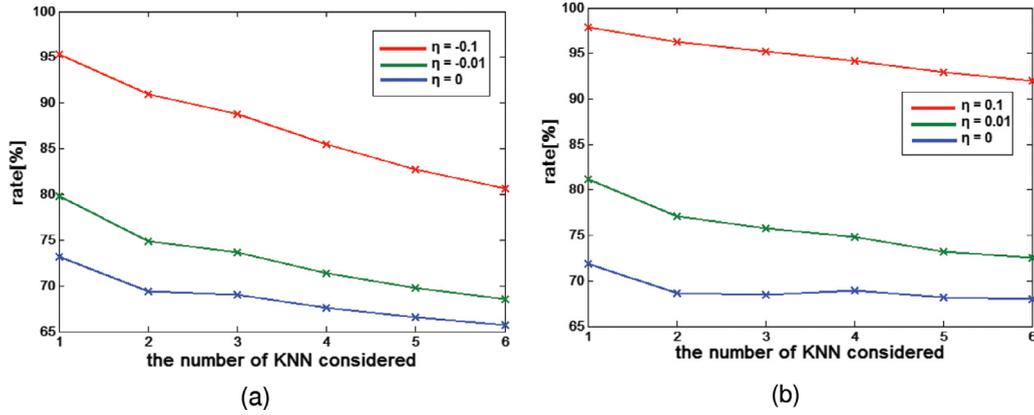


Fig. 3. (a) The statistical distributions of the distance between x_i and x_j ($x_i \in \mathcal{N}_k(q), x_j \in \mathcal{N}_k(p_2)$) on the whole dataset. The percentage of $\epsilon_{ij} \geq 1$ is always high. (b) The statistical distributions of the distance between x_i and x_j ($x_i \in \mathcal{N}_k(q), x_j \in \mathcal{N}_k(p_3)$) on the whole dataset. The percentage of $\epsilon_{ij} \leq 1$ is also high.

dataset as

$$\ell = \frac{\sum_{q \in \mathcal{Q}} \sum_{p \in \mathcal{P}_2(q)} f(\epsilon_{qp})}{\sum_{q \in \mathcal{Q}} |\mathcal{P}_2(q)|}, \tag{2}$$

where \mathcal{Q} denotes the database, and $|\cdot|$ calculates the set size. The function f is defined as

$$f(x) = \begin{cases} 1 & x \geq 1 \\ 0 & \text{otherwise} \end{cases}. \tag{3}$$

As can be seen from Fig. 3(a), the percentage increases with the value of η decreasing, the value of k decreasing. Generally, the value of the percentage l is large.

A similar statistical observation is conducted in $\mathcal{P}_3(q)$. Here we set η to 0.1, 0.01 and 0, and compute the percentage of $\epsilon_{ij} \leq 1$ on the whole dataset. Fig. 3(b) shows the statistical results, and find that the percentage is also high. As for \mathcal{P}_1 and \mathcal{P}_4 , obviously most of the distances between x_i and x_j are around the original $d(q, p)$, although some extreme high or low value (noise) may exist.

It is easy for us to understand such phenomena. As we all know, ideally x_i ($x_i \in \mathcal{N}_k(q)$) and q are a same instance from human eyes, but viewed under different angle and light, or have different rotation, translation and scaling, or have common objects. When the original descriptor cannot give $d(q, p)$ a proper value for some reasons (e.g. the shortcoming of the descriptor in rotation invariance), the distance between x_i and x_j will probably take a corrected value, which makes us possible to use the contextual information to revise the original measure for accurate retrieval in the view of statistics.

3.2. The definition of neighbor set similarity

Before introducing Neighbor Set Similarity, we give some necessary definitions. Given two sets $\mathcal{A} = \{a_1, a_2, \dots, a_n\}$ and $\mathcal{B} = \{b_1, b_2, \dots, b_m\}$, the similarity between point a and point b is defined as $s(a, b)$, then the similarity between point a and set \mathcal{B} can be defined as

$$s^\circ(a, \mathcal{B}) = \frac{1}{|\mathcal{B}|} \sum_{j=1}^m s(a, b_j). \tag{4}$$

The similarity between set \mathcal{A} and set \mathcal{B} is defined as

$$S(\mathcal{A}, \mathcal{B}) = \frac{1}{|\mathcal{A}|} \sum_{i=1}^n s^\circ(a_i, \mathcal{B}) = \frac{1}{|\mathcal{A}| \times |\mathcal{B}|} \sum_{i=1}^n \sum_{j=1}^m s(a_i, b_j). \tag{5}$$

Through Eq. (5), we can easily define the similarity between $\mathcal{N}_k(q)$ and $\mathcal{N}_k(p)$, which is summarized in Eq. (6):

$$S(\mathcal{N}_k(q), \mathcal{N}_k(p)) = \frac{1}{k^2} \sum_{x_i \in \mathcal{N}_k(q)} \sum_{x_j \in \mathcal{N}_k(p)} s(x_i, x_j). \tag{6}$$

Obviously, the size of $\mathcal{N}_k(q)$ and $\mathcal{N}_k(p)$ is k . Then the Neighbor Set Similarity defined on q and p is

$$s_{NSS}(q, p) = S(\mathcal{N}_k(q), \mathcal{N}_k(p)). \tag{7}$$

An illustration of the definition of NSS is presented in Fig. 4. It is easy to find that we use a set-to-set similarity to replace a point-to-point similarity, which makes NSS superior to the traditional matching algorithms. NSS has the following properties:

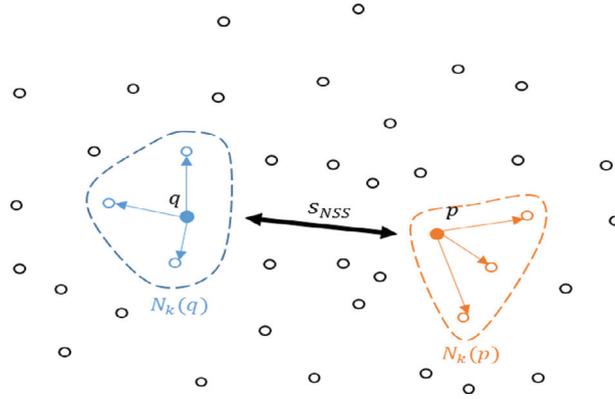


Fig. 4. The illustration of computing the Neighbor Set Similarity between two instances. The Neighbor Set Similarity between q and p , denoted by $s_{NSS}(q, p)$, is the average of all the pairwise similarities between two neighbor sets. Note that the neighbors of q are far from the neighbors of q in the manifold, then the Neighbor Set Similarity $s_{NSS}(q, p)$ is smaller than the original similarity $s(q, p)$.

1. **Robustness:** due to the fact that we convert a point-to-point similarity into a set-to-set similarity, NSS becomes robust to noise in view of statistics. Usually the two sets have tight relationship with q and p , and reveal the local distributions of the two instances properly.
2. **Locality:** NSS proves that more faithful information can be found by taking full use of the local distribution in the underlying manifold based on the original measure.
3. **Generality:** NSS is a generic algorithm, and can be built on top of any existing shape or image similarity measure. With the contextual information learned by NSS, higher retrieval precision can be obtained without many increasing processing or storage requirements.
4. **Symmetry:** the property of symmetry is obvious, i.e., $s_{NSS}(q, p) = s_{NSS}(p, q)$.
5. **Non-negativity:** the similarity value learned by NSS is restricted to the range $[0, 1]$ by the Gaussian kernel introduced in Section 3.4.

The proposed similarity is not a metric, since it does not obey the triangular inequality. Given three instances denoted by q , p_1 and p_2 respectively, it usually happens that q is far from both p_1 and p_2 , while p_1 is similar to p_2 in the similarity space. In this case, the similarity $s(q, p_1)$ and $s(q, p_2)$ is rather small, but $s(p_1, p_2)$ is very large. Hence we cannot get the triangular inequality defined as $s_{NSS}(q, p_1) + s_{NSS}(q, p_2) > s_{NSS}(p_1, p_2)$. The triplet relationship in the similarity space is well investigated in [4]. Although the proposed similarity is not a metric, it is proven to be effective in our specific scenario of retrieval.

In practice, since we consider query itself as the 0-nearest neighbor, NSS can enhance its robustness to noise due to the fact that if q and p are reciprocal nearest neighbors [26], the NSS between q and p will be further increased.

Compared with the popular diffusion process, NSS no longer needs an iterative process, which means that NSS is capable of large scale image retrieval to a certain extent. We will analysis the time complexity of NSS theoretically later, and show that NSS is not only more precise, but also faster than diffusion process.

The definition of NSS may look like the Hausdorff distance, which is widely used in object matching. Hausdorff distance between two sets \mathcal{A} and \mathcal{B} is to show the greatest of all distances from the point in set \mathcal{A} to the closet point in set \mathcal{B} :

$$D(\mathcal{A}, \mathcal{B}) = \max_{a \in \mathcal{A}} \min_{b \in \mathcal{B}} d(a, b). \quad (8)$$

Some variants of the standard Hausdorff distance are also proposed, such as partial Hausdorff distance and modified Hausdorff distance [43]. The NSS can be regarded as a variant of standard Hausdorff distance, but differs from it mainly in two aspects: (1) NSS is the definition of similarity between two sets, while Hausdorff distance defines the dissimilarity. (2) We replace the relationship between a point and a set defined by the operation of *min* in the Hausdorff distance by the operation of *mean*, which makes NSS more robust to noise in the task of retrieval. In fact, we have conducted lots of experiments about different variants of Hausdorff distance to solve the problem of retrieval, including the definition of Eq. (5) used in NSS. The experimental results prove that averaging all the similarities between two neighbor sets yields a more precise and more stable retrieval performance compared with other definitions.

Meanwhile, considering that the context of an instance is simply defined by its k -nearest neighbors, we also tried some other definitions in order to better reveal the local distribution of the data manifold. For example, Dominant Neighbors (DN) introduced in [38] were adopted to replace k -nearest neighbors. Although it brings a slight boost in the retrieval performance, the largely increased time consumption is intolerable. Moreover, different weight values were assigned to different instances in $\mathcal{N}_k(q)$ and $\mathcal{N}_k(p)$. It may be helpful to reduce the effect of noise, but more empirical parameter will be added, which makes the algorithm more complex. Notice that the only parameter should be determined empirically is k in the k NN according to the definition of NSS after the similarity between instances is given. Indeed, we finally give a, as simple as enough, model to capture the geometry

of the underlying manifold, because a good algorithm should be clear, effective, robust, and easy to understand. Besides, NSS is also a generic model, based on which many methods can improve their performances further.

3.3. Neighbor set similarity with more than two measures

As discussed above, different similarity measures may attach their emphasis to different aspects of an instance. In most cases, they are complementary to each other. In this section, we propose a novel method to combine different similarity measures inspired by co-training [5].

We take NSS with $m = 2$ input measures as example (and extension to $m > 2$ is similar). Given a set of instances $\mathcal{X} = \{x_1, x_2, \dots, x_n\}$ and a pair of similarity measures (α, β) , the similarity between q and p under the measure α is represented by $s_\alpha(q, p)$. The set of k -nearest neighbors of q and p are denoted by $\mathcal{N}_{k,\alpha}(q)$ and $\mathcal{N}_{k,\alpha}(p)$ respectively when using measure α as the descriptor. The similarity between $\mathcal{N}_{k,\alpha}(q)$ and $\mathcal{N}_{k,\beta}(p)$ under the measure α is defined as

$$S_\alpha(\mathcal{N}_{k,\alpha}(q), \mathcal{N}_{k,\beta}(p)) = \frac{1}{k^2} \sum_{x_i \in \mathcal{N}_{k,\alpha}(q)} \sum_{x_j \in \mathcal{N}_{k,\beta}(p)} s_\alpha(x_i, x_j). \quad (9)$$

Then the Neighbor Set Similarity between q and p with $m = 2$ measures is defined as follows:

$$s_{NSS}(q, p) = \text{mean}\{S_\alpha(\mathcal{N}_{k,\alpha}(q), \mathcal{N}_{k,\beta}(p)), S_\beta(\mathcal{N}_{k,\beta}(q), \mathcal{N}_{k,\alpha}(p))\}. \quad (10)$$

Similarly, we do not need an iterative process to guarantee the retrieval precision. Only one step to fuse two measures has already yield a better performance than other state-of-art algorithms. We will experimentally prove the discriminative power of fusion process based on NSS in Section 5.

3.4. The similarity matrix

In shape and image retrieval, a distance measure is often defined, e.g. [3,15]. Let $D = [d_{ij}]_{n \times n}$ be the distance matrix computed by a certain distance measure, and a kernel function can be adopted to convert it to a similarity matrix $S = [s_{ij}]_{n \times n}$. In this paper, Gaussian kernel defined as

$$s(q, p) = \exp(-d^2(q, p)/\delta_{qp}^2) \quad (11)$$

is utilized. The value of the kernel δ_{qp} can be determined by studying the local statistics of the neighbors of q and p . δ_{qp} is usually defined as

$$\delta_{qp} = \alpha \cdot \text{mean}(\text{knn_dist}(q), \text{knn_dist}(p)), \quad (12)$$

where $\text{mean}(\text{knn_dist}(q), \text{knn_dist}(p))$ represents the mean distance of the k -nearest neighbor distances of q and p . Both k and α are determined empirically. We set α to 0.33 according to the three-sigma rule.

4. Relation to diffusion process

Diffusion process on affinity graph has been proven its ability to significantly improve the retrieval precision. In this section, we compare NSS with a popular diffusion process called Local Constrained Diffusion Process (LCDP) [37] briefly. LCDP is defined as

$$P_{KK}^{t+1}(q, p) = \sum_{i \in \text{kNN}(q), j \in \text{kNN}(p)} P(q, x_i) P_{KK}^t(x_i, x_j) P(x_j, p), \quad (13)$$

where $P^t(q, p)$ represents the transition probability from node q to node p after t times iteration. The essence of LCDP is finding a path between k NNs of q and k NNs of p . $P(q, p)$ is high if all the paths between points in $kNN(q)$ and $kNN(p)$ are short.

Some common parts are easy to be found when comparing Eqs. (13) and (6) that they both use the contextual information to define the relationship between q and p . The difference is that LCDP uses the operation of multiplication in order to compute the transition probability from q to p , while NSS uses the operation of addition in order to compute the statistical mean distance between k NNs of q and k NNs of p . In fact, the two ways are constitutionally the same.

However, diffusion process, e.g. LCDP, can not undertake the task for efficient retrieval, because the time complexity of most diffusion processes is at least $O(n^3)$, which is unbearable when the size of database becomes bigger (we assume the database size is n). Furthermore, they require matrix multiplication operations that is usually computed more slowly compared with the operation of addition. In comparison, NSS only takes the similarity between the elements in $\mathcal{N}_k(q)$ and $\mathcal{N}_k(p)$, which computes k^2 times. When considering that we need to compute NSS for each two instances in the database, the time complexity of NSS is $O(k^2 \times n^2)$. In practice, the value of k is much smaller than the value of n . As a result the time complexity of NSS is only $O(n^2)$.

What is more important is that NSS does not, but diffusion process does, need an iterative process to get a higher precision, NSS only needs one step to give the final retrieval result, also a better result. In practice, we can speed up the retrieval process with parallel computing, i.e., more computing cores can be used to compute NSS between instances, because the computing of NSS can be divided into lots of subtasks with small scale.

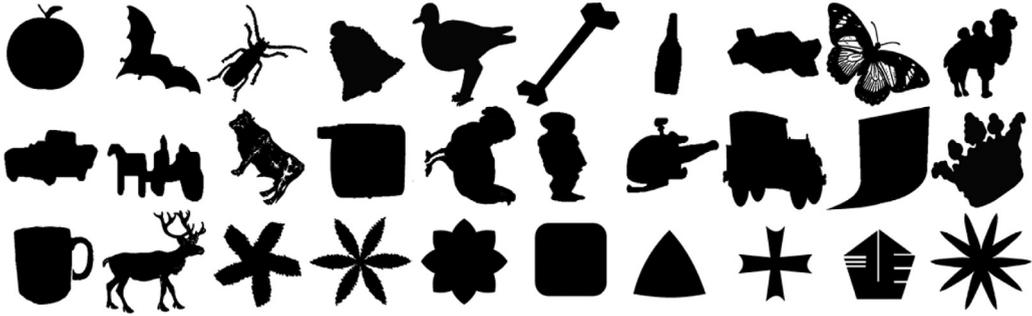


Fig. 5. Some typical shapes from the MPEG-7 dataset.

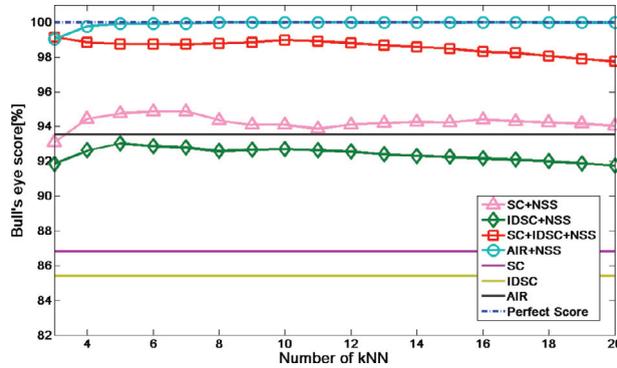


Fig. 6. The bull's eye scores on MPEG-7 dataset with different parameter setting. The dotted blue line represents the bull's eye score of 100%. Note that AIR+NSS always achieves the perfect results after $k > 7$.

5. Experiments

In this section, we use MPEG-7 shape dataset [14], Nister and Stewenius image dataset [22] and ORL face dataset [27] to demonstrate the validity of NSS. We compare NSS with some related methods, such as Locally Constrained Diffusion Process [37], Tensor Product Graph [38] and so on. All the experiments are carried out on a personal computer with an Intel(R) Core(TM) i5-2320 K CPU (3.00 GHz) and 12GB RAM memory. The experimental results show that NSS can significantly improve the retrieval performance over other state-of-art diffusion algorithms.

5.1. Shape retrieval

Our proposed algorithm is first tested for shape retrieval on a widely-used MPEG-7 shape dataset. It consists of 1400 silhouette images grouped into 70 classes, and each class has 20 binary shapes. Some exemplars are shown in Fig. 5. The “bull’s eye test” is often used to evaluate the retrieval precision on the MPEG-7 dataset. Each shape in the dataset is compared to all other shapes, and the shapes from the same class among the 40 most similar ones are considered as correct retrieval results. The bull’s eye score is the ratio of the total number of correct retrieval shapes to the highest number possible 1400×20 , which means the best possible score is 100%.

In order to show the effectiveness of NSS, we use three popular shape descriptors, Shape Context (SC) [3], Inner Distance Shape Context (IDSC) [15] and Articulation-Invariant Representation (AIR) [11], as the original input measures to compute the pairwise distances between shapes. The performances of SC, IDSC and AIR are 86.80%, 85.40% and 93.67% respectively.

As discussed above, the parameter k in the definition of kNN should be determined empirically, and Fig. 6 presents the influence on bull’s eye score of different values of k . It can be drawn that generally, the performance of NSS improves when the neighborhood size increasing, and deteriorates slowly after the neighborhood size is 6. From the plots, we also find that NSS is insensitive to parameter tuning. For example, all the retrieval scores of NSS with AIR as the input measures are above 99%. Perfect score can be obtained by NSS with AIR when $k > 7$.

In addition, we report the percent gain for each class in MPEG-7 dataset obtained by NSS compared with the original measure. As shown in Fig. 7(a), the bull’s eye scores in the majority of classes are improved by NSS compared to the original measure SC by different ratios. In five shape classes, the percent gain is above 50%, and the largest gain is 74.50%, which verifies the positive effect of NSS firmly. As for the statistical results presented in Fig. 7(b), similar phenomenon can be observed that NSS can boost the original IDSC by learning the contextual information. However, we also find that in two shape classes, NSS brings down the precision by a small percentage on the contrary. Our interpretation is that the shape class composed of spoons (63rd class in the

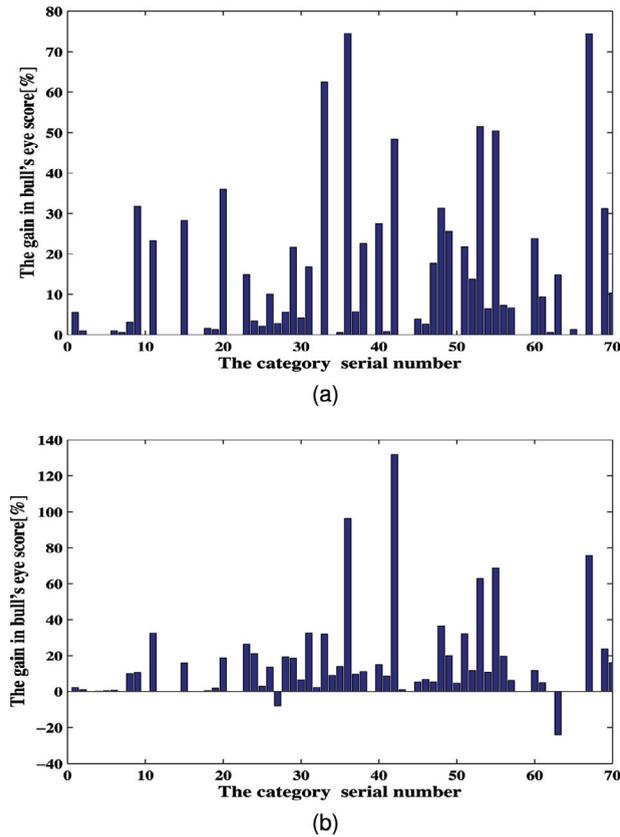


Fig. 7. The percent gain in bull's eye retrieval rates for each class in the MPEG-7 dataset of SC (a) and IDSC (b).

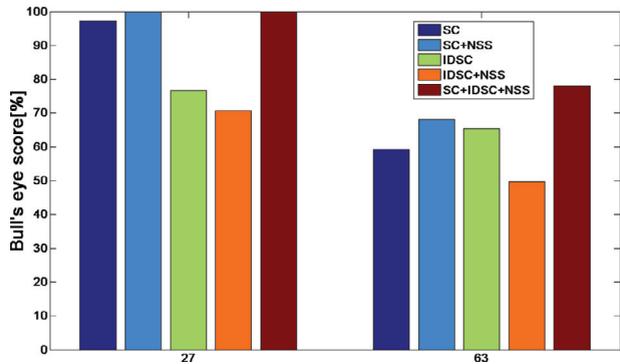


Fig. 8. The illustration for the complementarity of SC and IDSC. The number 27 and 63 represent the shape class composed of spoon and device0-1 from MPEG-7 dataset.

histogram of Fig. 7(b)) is similar to the guitar class under the measure of IDSC, which leads to the fact that an excess of noise is included in the contextual information defined by kNN. Many similar algorithms always need as precise as enough contextual information, including NSS. However, NSS is more robust compared with other algorithms, when observing that the abnormal performance of NSS depicted in Fig. 7(b) is limited and tiny. On the other hand, it also inspires us to employ more complementary similarity measures (*i.e.*, SC) to make up the weakness of IDSC in this situation.

In order to visualize the complementarity that exists between SC and IDSC, we compare the bull's eye score of the two shape classes of spoon and device 0-1 measured by different algorithms in Fig. 8. It can be drawn from the histogram that when the performance of IDSC is poor, and NSS with IDSC as the input similarity measure cannot correct the false retrieval results due to the poor recognition capability of IDSC under the specific circumstances. SC and SC in conjunction with NSS generate more precise retrieval results. Thus, fusing SC and IDSC is natural to achieve a better performance at least in these two shape classes. Experimental results in Fig. 8 show that NSS with SC and IDSC as the input measures significantly improves the retrieval precision in the two shape classes, in which IDSC has a poor performance. In Fig. 6, we also observe that the retrieval precision

Table 1

Bull's eye score and average retrieval time of different algorithms on MPEG-7 dataset. Note that DDGM [29], IDSC [15], SC [3], ASC [16], Height functions [33] and AIR [11] are merely shape descriptors without any re-ranking algorithms, so their computational cost for re-ranking are ignored.

Algorithm	Shape descriptor	Bull's eye score (%)	Time
–	DDGM [29]	80.03	–
–	IDSC	85.40	–
–	SC	86.80	–
–	ASC [16]	88.39	–
–	Height functions [33]	86.66	–
CDM [12]	IDSC	88.30	34.42 ms
Hausdorff distance	SC	90.71	0.74 ms
LP [36]	IDSC	91.61	0.91 s
Pairwise recom. [23]	IDSC	92.21	35.71 ms
Modified Hausdorff distance	SC	92.33	0.71 ms
LCDP [37]	IDSC	93.32	55.14 ms
SSP [34]	IDSC	93.35	2.95 s
–	AIR [11]	93.67	–
NSS	SC	94.89	0.67 ms
TPG [39]	ASC	96.47	0.28 s
Co-trans. [1]	SC+IDSC	97.72	1.79 s
NSS	SC+IDCS	99.15	0.71 ms
TPG	AIR	99.99	0.28 s
NSS	AIR	100	0.67 ms

of NSS fusing SC and IDSC on the whole dataset is much better than NSS only using one type of similarity measure, which demonstrates the validity of our proposed fusing method.

We also evaluate NSS in comparison with other state-of-art algorithms in Table 1. We can find that NSS outperforms most compared algorithms, including some popular diffusion based algorithms. The performances of standard Hausdorff distance and modified Hausdorff distance defined between $\mathcal{N}_k(q)$ and $\mathcal{N}_k(p)$ are inferior to NSS. It indicates that averaging all the similarities between two neighborhood sets is a more proper behavior in retrieval task. Although the difference between the best score of TPG and NSS is tiny, NSS achieves the perfect score of 100%, which makes a big difference. As we will show next, NSS obtains perfect score with higher computing efficiency compared with TPG.

We have discussed above that the time complexity of many diffusion processes is at least $o(n^3)$, which makes diffusion process improper to deal with large scale retrieval. What makes things worse is that most diffusion processes need an iterative procedure to guarantee the retrieval precision. The operation of matrix multiplication is more time-consuming compared with the operation of addition. In order to show the potential of NSS for the large scale data retrieval, a comparison about the running time for retrieving a certain query is also presented in Table 1. We only focus on the time cost from the moment that the original pairwise similarity values are given to the end of retrieval, since NSS is a post-progressing algorithm focusing on re-ranking procedure. The time cost for matching each query of NSS is only 0.67 ms, which is the most efficient one among all the compared re-ranking algorithms. The time cost of two baseline methods, LCDP and TPG, are nearly 82 times and 425 times as long as that of NSS. Obviously, NSS is much more computationally efficient than TPG and LCDP. The running time of TPG is much longer because of the high time cost for computing Dominant Neighbors. We can draw a conclusion that NSS is more capable of dealing with large scale image retrieval to a certain extent compared with other diffusion based algorithms.

5.2. Image retrieval

In this section, we present the performance of the proposed approach for image retrieval, and we select the Nister and Stewenius (N-S) dataset. The dataset is also known as the UKBench dataset. N-S dataset contains 2550 objects, and each takes four different viewpoints. Hence, there are 10,200 images in N-S dataset in total. Some sample images are shown in Fig. 9. The retrieval precision is measured by the average number of correct images in the top 4 images returned. Thus, the best score is 4. There are only 4 images in each class, which makes the dataset very challenging for any manifold learning method. However, the experimental results prove that NSS also performs well in such a challenging dataset.

Here we use a local descriptor and a global descriptor as the original measure to compute the pairwise distance values. The local descriptor is extracted with the Hessian-affine detector [21] and described by SIFT [20]. A visual vocabulary is learned using the K-means algorithm on the extracted SIFT descriptors, and Bag of Words (BoW) is used to encode these local descriptors to obtain a visual histogram representing the image. GIST [28], which employs a visual attention model to combine global color, intensity and orientation features, is selected as the global descriptor. The N-S scores of the two methods are 3.41 and 2.88 respectively.

We compare NSS with other diffusion based methods and a recently proposed learning algorithm called Contextual Dissimilarity Measure (CDM) [12]. CDM follows a different principle from diffusion process. CDM is motivated by an observation that a good ranking is usually not symmetrical in image search. CDM defines two images to be similar when they both obtain a



Fig. 9. Some sample images from the N-S dataset.



Fig. 10. Some example face images from the ORL face dataset.

Table 2
Comparison of scores on Nister and Stewenius dataset.

Algorithm	N-S score	Algorithm	N-S score
VOC [35]	3.54	Co-transduction [1]	3.66
CDM [12]	3.57	CrDP [30]	3.68
LCDP [37]	3.58	CSFS [32]	3.69
TPG [39]	3.61	Graph rank [44]	3.76
SIFT+NSS	3.66	Graph density [44]	3.77
GIST+NSS	3.69	SIFT+GIST+NSS	3.80

good ranking position when using each other as a query. It can improve the retrieval performance especially when the original similarity measure is given by BoW model.

The retrieval results are shown in Table 2. The fact that our proposed method improves the baseline of GIST significantly (from 2.88 to 3.69) proves the effectiveness of NSS. As for two baseline methods, TPG [39] reports N-S score 3.61, but the performance of LCDP [37] is not accessible. We report the result of LCDP using our implementation of SIFT descriptors. As the table shows, LCDP improves the performance of SIFT from 3.41 to 3.58. The score of NSS fusing SIFT and GIST is 3.80, which is the state-of-the-art to the best of our knowledge now.

5.3. Face retrieval

We also conduct an experiment of face retrieval on ORL dataset. ORL is a face dataset with 40 subjects, and each subject has 10 grayscale images, where pose, expression and illumination are different. Some example face images on ORL face dataset are presented in Fig. 10. The bull's eye score that considering 15 closest neighbors is used to evaluate the retrieval result.

In order to show the advantage of NSS and ensure a fair comparison with other algorithms, we adopt the same distance matrix offered by Donoser and Bischof in [9]. Specifically, each image is down-sampled and then normalized to 0-mean and 1-variance. Euclidean distance is adopted to compute the pairwise distances between the vectorized representations. The baseline of ORL dataset is 62.35%, and the experimental results presented in Table 3 show that our proposed method achieves a higher retrieval score compared with diffusion processes summarized in [9], including LCDP [37] and TPG [39]. We refer the readers to [9] for more details about the generic diffusion process framework.

Table 3
Results on the ORL dataset.

Algorithm	Score (%)
SD [31]	71.68
TPG [38]	73.90
LCDP [37]	74.25
Generic diffusion [9]	77.42
NSS	77.98

Besides, GIST descriptor is also extracted to evaluate the performance of NSS, which improves the retrieval precision to **96.65%**.

6. Conclusion

In this paper, we present a simple yet effective re-ranking method called Neighbor Set Similarity. NSS makes full use of contextual information from the original measure, thus obtains a more precise similarity measure. In addition, based on NSS, we propose a powerful fusion process to fuse two different complementary similarity measures, and achieve more faithful information.

Compared with diffusion process, NSS has its advantages mainly in three aspects: (1) More precise. Extensive experiments prove that NSS performs better than diffusion process. NSS gains its robustness to noise in the view of statistics; (2) Faster. The time complexity of diffusion process is larger than that of NSS, and it also needs an iterative procedure to guarantee its validity. However, NSS always returns a precise enough retrieval result efficiently. (3) Practicability. Due to the property of (1)(2), NSS is proper for commercial purpose in many search engines for real-time applications. Moreover, NSS can be done with parallel computing, which makes it more suitable for practical applications.

Acknowledgment

This work was primarily supported by National Natural Science Foundation of China (NSFC) (no. 61222308), and in part by NSFC (no. 61173120), Program for New Century Excellent Talents in University (no. NCET-12-0217). The authors would like to thank the anonymous reviewers for their valuable suggestions.

References

- [1] X. Bai, B. Wang, C. Yao, W. Liu, Z. Tu, Co-transduction for shape retrieval, *Trans. Process.* 21 (5) (2012) 2747–2757.
- [2] X. Bai, X. Yang, L.J. Latecki, W. Liu, Z. Tu, Learning context-sensitive shape similarity by graph transduction, *Trans. Pattern Anal. Mach. Intell.* 32 (5) (2010) 861–874.
- [3] S. Belongie, J. Malik, J. Puzicha, Shape matching and object recognition using shape contexts, *Trans. Pattern Anal. Mach. Intell.* 24 (4) (2002) 509–522.
- [4] I. Ben-Shalom, N. Levy, L. Wolf, N. Dershowitz, A. Ben-Shalom, R. Shweka, Y. Choueka, T. Hazan, Y. Bar, Congruency-based reranking, in: *Proceedings of the Computer Vision and Pattern Recognition*, 2014.
- [5] A. Blum, T. Mitchell, Combining labeled and unlabeled data with co-training, in: *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*, 1998.
- [6] S. Changpinyo, K. Liu, F. Sha, Similarity component analysis, in: *Proceedings of the Neural Information Processing Systems*, 2013.
- [7] R.R. Coifman, S. Lafon, Diffusion maps, *Appl. Comput. Harmonic Anal.* 21 (1) (2006) 5–30.
- [8] M.R. Daliri, V. Torre, Robust symbolic representation for shape recognition and retrieval, *Pattern Recognit.* 41 (5) (2008) 1782–1798.
- [9] M. Donoser, H. Bischof, Diffusion processes for retrieval revisited, in: *Proceedings of the Computer Vision and Pattern Recognition*, 2013.
- [10] C. Gong, D. Tao, K. Fu, J. Yang, Fick's law assisted propagation for semisupervised learning, *Trans. Neural Netw. Learn. Syst.* (2014).
- [11] R. Gopalan, P. Turaga, R. Chellappa, Articulation-invariant representation of non-planar shapes, in: *Proceedings of the European Conference on Computer Vision*, 2010.
- [12] H. Jegou, C. Schmid, H. Harzallah, J. Verbeek, Accurate image search using the contextual dissimilarity measure, *Trans. Pattern Anal. Mach. Intell.* 32 (1) (2010) 2–11.
- [13] J. Jiang, B. Wang, Z. Tu, Unsupervised metric learning by self-smoothing operator, in: *Proceedings of the International Conference on Computer Vision*, 2011.
- [14] L.J. Latecki, R. Lakamper, T. Eckhardt, Shape descriptors for non-rigid shapes with a single closed contour, in: *Proceedings of the Computer Vision and Pattern Recognition*, 2000.
- [15] H. Ling, D.W. Jacobs, Shape classification using the inner-distance, *Trans. Pattern Anal. Mach. Intell.* 29 (2) (2007) 286–299.
- [16] H. Ling, X. Yang, L.J. Latecki, Balancing deformability and discriminability for shape matching, in: *Proceedings of the European Conference on Computer Vision*, 2010.
- [17] M. Liu, B. Vemuri, S.-i. Amari, F. Nielsen, Shape retrieval using hierarchical total Bregman soft clustering, *Trans. Pattern Anal. Mach. Intell.* 34 (12) (2012) 2047–2419.
- [18] W. Liu, D. Tao, Multiview hessian regularization for image annotation, *Trans. Image Process.* 22 (7) (2013) 2676–2687.
- [19] W. Liu, D. Tao, J. Cheng, Y. Tang, Multiview hessian discriminative sparse coding for image annotation, *Comput. Vis. Image Underst.* 118 (2014) 50–60.
- [20] D.G. Lowe, Distinctive image features from scale-invariant keypoints, *Int. J. Comput. Vis.* 60 (2) (2004) 91–110.
- [21] K. Mikolajczyk, C. Schmid, Scale and affine invariant interest point detectors, *Int. J. Comput. Vis.* 60 (1) (2004) 63–86.
- [22] D. Nister, H. Stewenius, Scalable recognition with a vocabulary tree, in: *Proceedings of the Computer Vision and Pattern Recognition*, 2006.
- [23] D.C.G.e. Pedronette, R. da S Torres, Exploiting pairwise recommendation and clustering strategies for image re-ranking, *Inf. Sci.* 207 (2012) 19–34.
- [24] S.-H. Peng, D.-H. Kim, S.-L. Lee, C.-W. Chung, A visual shape descriptor using sectors and shape context of contour lines, *Inf. Sci.* 180 (16) (2010) 2925–2939.
- [25] V. Premachandran, R. Kakarala, Consensus of k-NNS for robust neighborhood selection on graph-based manifolds, in: *Proceedings of the Computer Vision and Pattern Recognition*, 2013.
- [26] D. Qin, S. Gammeter, L. Bossard, T. Quack, L. Van Gool, Hello neighbor: accurate object retrieval with k-reciprocal nearest neighbors, in: *Proceedings of the Computer Vision and Pattern Recognition*, 2011.

- [27] F.S. Samaria, A.C. Harter, Parameterisation of a stochastic model for human face identification, in: Proceedings of the Second IEEE Workshop on Applications of Computer Vision, 1994.
- [28] C. Siagian, L. Itti, Rapid biologically-inspired scene classification using features shared with visual attention, *Trans. Pattern Anal. Mach. Intell.* 29 (2) (2007) 300–312.
- [29] Z. Tu, A.L. Yuille, Shape matching and recognition using generative models and informative features, in: Proceedings of the European Conference on Computer Vision, 2004.
- [30] B. Wang, J. Jiang, W. Wang, Z.-H. Zhou, Z. Tu, Unsupervised metric fusion by cross diffusion, in: Proceedings of the Computer Vision and Pattern Recognition, 2012.
- [31] B. Wang, Z. Tu, Affinity learning via self-diffusion for image segmentation and clustering, in: Proceedings of the Computer Vision and Pattern Recognition, 2012.
- [32] C. Wang, L. He, Y. Zhu, W. Liu, Learning context sensitive similarity measure on pair fusion graph, in: Proceedings of the International Conference on Image Processing, 2013.
- [33] J. Wang, X. Bai, X. You, W. Liu, L.J. Latecki, Shape matching and classification using height functions, *Pattern Recognit. Lett.* 33 (2) (2012) 134–143.
- [34] J. Wang, Y. Li, X. Bai, Y. Zhang, C. Wang, N. Tang, Learning context-sensitive similarity by shortest path propagation, *Pattern Recognit.* 44 (10) (2011) 2367–2374.
- [35] X. Wang, M. Yang, T. Cour, S. Zhu, K. Yu, T.X. Han, Contextual weighting for vocabulary tree based image retrieval, in: Proceedings of the Computer Vision and Pattern Recognition, 2011.
- [36] X. Yang, X. Bai, L.J. Latecki, Z. Tu, Improving shape retrieval by learning graph transduction, in: Proceedings of the European Conference on Computer Vision, 2008.
- [37] X. Yang, S. Koknar-Tezel, L.J. Latecki, Locally constrained diffusion process on locally densified distance spaces with applications to shape retrieval, in: Proceedings of the Computer Vision and Pattern Recognition, 2009.
- [38] X. Yang, L.J. Latecki, Affinity learning on a tensor product graph with applications to shape and image retrieval, in: Proceedings of the Computer Vision and Pattern Recognition, 2011.
- [39] X. Yang, L. Prasad, L.J. Latecki, Affinity learning with diffusion on tensor product graph, *Trans. Pattern Anal. Mach. Intell. TPAMI* (2013).
- [40] J. Yu, Y. Rui, B. Chen, Exploiting click constraints and multi-view features for image re-ranking, *Trans. Multimed. TMM* 16 (1) (2014) 159–168.
- [41] J. Yu, Y. Rui, D. Tao, Click prediction for web image reranking using multimodal sparse coding, *Trans. Image Process. TIP* 23 (5) (2014) 2019–2032.
- [42] J. Yu, D. Tao, M. Wang, Y. Rui, Learning to rank using user clicks and visual features for image retrieval, *Trans. Cybern.* 45 (4) (2015) 767–779.
- [43] M.-L. Zhang, Z.-H. Zhou, Multi-instance clustering with applications to multi-instance prediction, *Appl. Intell.* 31 (1) (2009) 47–68.
- [44] S. Zhang, M. Yang, T. Cour, K. Yu, D.N. Metaxas, Query specific fusion for image retrieval, in: Proceedings of the ECCV, 2012.
- [45] L. Zheng, S. Wang, Z. Liu, Q. Tian, Packing and padding: coupled multi-index for accurate image retrieval, in: Proceedings of the Computer Vision and Pattern Recognition, 2014.
- [46] L. Zheng, S. Wang, Q. Tian, Coupled binary embedding for large-scale image retrieval, *Trans. Image Process. TIP* 23 (8) (2014) 3368–3380.
- [47] D. Zhou, J. Weston, A. Gretton, O. Bousquet, B. Schölkopf, Ranking on data manifolds, in: Proceedings of the NIPS, 2004.
- [48] Z.-H. Zhou, M. Li, Tri-training: exploiting unlabeled data using three classifiers, *Trans. Knowl. Data Eng.* 17 (11) (2005) 1529–1541.
- [49] Z.-H. Zhou, M. Li, Semi-supervised regression with co-training, in: Proceedings of the International Joint Conferences on Artificial Intelligence, 2005.
- [50] X. Zhu, J. Lafferty, R. Rosenfeld, Semi-supervised learning with graphs, Ph.D. thesis. Carnegie Mellon University, Language Technologies Institute, School of Computer Science, 2005.