Pattern Recognition 83 (2018) 353–364

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

Improving context-sensitive similarity via smooth neighborhood for object retrieval

Song Bai[a], Shaoyan Sun[b], Xiang Bai[a,*], Zhaoxiang Zhang[c], Qi Tian[d]

[a] Huazhong University of Science and Technology, China
[b] University of Science and Technology of China, China
[c] CAS Center for Excellence in Brain Science and Intelligence Technology, CASIA, China
[d] University of Texas at San Antonio, United States

ARTICLE INFO

Article history:
Received 30 October 2017
Revised 15 March 2018
Accepted 1 June 2018
Available online 22 June 2018

Keywords:
Object retrieval
Context-sensitive similarity
3D shape
Re-ranking
Rank aggregation

ABSTRACT

Due to the ability of capturing the geometry structure of data manifold, context-sensitive similarity has demonstrated impressive performances in the retrieval task. The key idea of context-sensitive similarity is that the similarity between two data points can be more reliably estimated with the local context of other points in the affinity graph. Therefore, neighborhood selection is a crucial factor for those algorithms, which affects the performance dramatically. In this paper, we propose a new algorithm called Smooth Neighborhood (SN) that mines the neighborhood structure to satisfy the manifold assumption. By doing so, nearby points on the underlying manifold are guaranteed to yield similar neighbors as much as possible. Moreover, SN is adjusted to tackle multiple affinity graphs by imposing a weight learning paradigm, and this is the primary difference compared with related works which are only applicable with one affinity graph. Finally, we integrate SN with Sparse Contextual Activation (SCA), a representative context-sensitive similarity proposed recently. Extensive experimental results and comparisons manifest that with the neighborhood structure generated by SN, the proposed framework can yield state-of-the-art performances on shape retrieval, image retrieval and 3D model retrieval.

1. Introduction

Object retrieval [1] is an important topic in pattern recognition [2], computer vision [3], multimedia computing [4,5] and machine learning, which has been investigated for decades. A typical retrieval system receives a query data as its input, and outputs the searching results which are expected to be visually similar to the given query. Therefore, the crucial issue in object retrieval is to define a reliable similarity between the query object and the database elements. In most cases, visual descriptors that are robust to common deformations (e.g., rotation, occlusion, illumination) are designed to assign each object a vectorial representation. Then, the pairwise matching between objects can be done in the Euclidean distance.

In recent years, context-sensitive similarity [6,7] has attracted much attention due to its superior performances in object retrieval, which does not solely rely on the pairwise matching. These approaches have a very diverse nomenclature, such as contextual dissimilarity measure [8], graph transduction [9–11], affinity learning [12,13], ranking list comparison [2,14,15], re-ranking [16–18]. However, the inherent principles of most those algorithms are almost the same, that is, the similarity between two data points can be more accurately measured by taking the underlying manifold structure into account. In order to specify the differences among them systematically, Donoser and Bischof, [19] provide a generic framework called Diffusion Process and a thorough comparison of most aforementioned algorithms experimentally. Diffusion process is usually operated on an affinity graph, with the nodes representing data points and the edge weights denoting the pairwise similarities between two adjacent nodes. In fact, the affinity graph defines a data manifold implicitly, then the similarities are diffused along the geodesic path of the manifold.

As one of the most important conclusions quoted from Donoser and Bischof [19], it is crucial to constrain the diffusion process locally, since diffusion process is susceptible to noise edges in the affinity graph. The experimental observation supports the “locality” assumption in manifold learning [20], that each data point and its neighbors lie on a linear patch of the manifold. It means that only quite short distances are reliable since they tend to associate

[*] Corresponding author.
E-mail addresses: songbai.site@gmail.com (S. Bai), sunshy@mail.ustc.edu.cn (S. Sun), xbai@hust.edu.cn (X. Bai), zhaoxiang.zhang@ia.ac.cn (Z. Zhang), qi.tian@utsa.edu (Q. Tian).

https://doi.org/10.1016/j.patcog.2018.06.001
0031-3203/© 2018 Elsevier Ltd. All rights reserved.

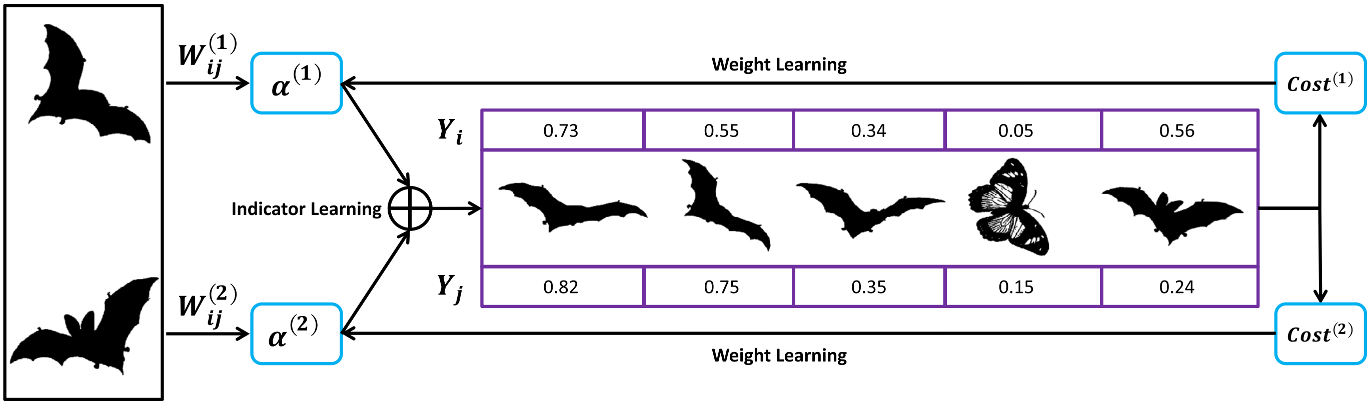

Fig. 1. The illustration of the proposed method with two similarity measures. For each pair of objects (x_i, x_j) having similarity $W_{ij}^{(v)}$ ($v = 1, 2$), we learn their indicator functions on neighbor selections, which is constrained by the similarities. Then, the resulted costs can be used, in turn, to learn the weights of those similarities.

with short geodesic distances along the data manifold. As a consequence, the nodes that diffusion process selects to spread the similarities on the affinity graph are usually the neighbors of the query which have large similarities (or small dissimilarities) with it. Hence, it is of great importance to construct robust neighborhood structures so that diffusion process can be performed in a proper way.

The simplest way to establish the neighborhood structure is k-Nearest Neighbor (kNN) rule. Given a certain query, kNN rule selects K nodes with the largest edge weights to the query as its neighborhood. Some variants of kNN are also proposed, such as ϵ-neighbors, symmetric kNN, Mutual kNN [21] (also named as reciprocal kNN in [22,23]). As extensively proven in [21], kNN is prone to including noise edges and nodes, thus leading to unsatisfactory retrieval performances. To overcome its defect, Dominant Neighborhood (DN) is proposed in [12] based on the analysis of dominant sets, and Consensus of kNN (CN) is proposed in [24] by exploiting the consensus information of kNN.

However, although these neighborhood analysis algorithms are embedded into some variants of diffusion process, they themselves do not capture the geometry of the data manifold. That is to say, they cannot preserve the property of *local consistency* that nearby points on the manifold are guaranteed to yield the same neighbors. For example, it usually occurs that two points belong to the same dense cluster, while they have no common neighbors if kNN rule or DN is used. In context-based retrieval, this problem is first proposed in [25], and later emphasized again in [19]. Nevertheless, they only alleviate the problem to a certain extent by localizing the diffusion process using kNN on both sides (query side and database side), and do not intend to tackle the problem seriously.

Moreover, previous works are only applicable with one affinity graph. When multiple affinity graphs are given, it becomes more challenging to accurately construct the neighborhood structures. Therein, the difficulties lie in two aspects. First, it is problematic to determine the weights of affinity graphs, which can distinguish the discriminative capacity of different similarity measures. Second, it is hard to aggregate the neighborhood structures generated with different affinity graphs, especially considering that retrieval is usually defined as an unsupervised task without prior knowledge. Of course, one can simply use a linear combination of multiple affinity graphs with equal weights. However, as demonstrated in our experiments, it is a suboptimal solution since the complementary nature among multiple similarities is neglected.

In this paper along with its earlier conference version [26], we propose an algorithm called Smooth Neighborhood (SN) specifically for neighborhood structure mining. As illustrated in Fig. 1, the motivation of SN is that the indicator functions can be defined to reveal the behavior of neighbor selection on affinity graphs thus yielding a selecting cost for each graph, then the resulted costs can be used, in turn, to learn the weights of those affinity graphs in an unsupervised manner. Apart from related works, our primary contributions can be divided into three parts:

1. SN enables the neighbor selection to vary smoothly with respect to the local geometry of the data manifold, thus the input similarity can be sufficiently reflected in the behavior of neighbor selection.
2. SN is suitable to deal with more than one affinity graph. It learns a shared neighborhood structure and the importance of multiple affinity graphs in a unified framework. Therefore, the weight learning procedure and the neighborhood aggregation procedure can be done simultaneously.
3. Instead of using some heuristic rules that stem from empirical observations (e.g., Mutual kNN), we give a formal formulation to SN and derive an iterative solution to the optimization problem with proven convergence.

Compared with the conference version, the work (1) gives a deeper analysis on the motivation and the difference from relevant works; (2) supplements the properties of SN, such as the proof of convergence and convexity; (3) provides more thorough experimental evaluations with different types of data pattern, such as 3D model retrieval.

The rest of paper is organized as follows. In Section 2, we review some representative algorithms which have a close relationship with SN. The formulation and optimization of SN are given in Section 3. In Section 4, the effectiveness of SN is verified with thorough experiments and comparisons on shape retrieval, image retrieval and 3D model retrieval. Conclusions are given in Section 5.

2. Related work

Tremendous developments in context-sensitive similarities advance image and shape retrieval remarkably. A family of algorithms called diffusion process is proposed in the literature, such as Graph Transduction [9], Locally Constrained Diffusion Process (LCDP) [25], Locally Constrained Mixed Diffusion (LCMD) [27], Tensor Product Graph Diffusion (TPG) [12], Shortest Path Propagation (SSP) [10], Graph-PageRank [17], etc. In the survey paper [19], most of these approaches are elegantly summarized in a unified framework.

As shown in [19], a proper selection of neighbors ensures the diffusion process to work well in real cases. However, most variants of diffusion process use k-Nearest Neighbor (kNN) rule for its simplicity. Although [19] also uses kNN rule, it points out that it is still an open issue to select a reasonable local neighborhood. Related to this task, there are two representative algorithms in recent years, i.e., Dominant Neighborhood (DN) [12] and Consensus of kNN (CN) [24]. Here, we briefly review their motivations and formulations to better highlight the contribution of this work.

Given a collection of images $X = \{x_1, x_2, \dots, x_N\}$, we can construct an undirect graph $\mathcal{G} = (X, W)$, where the nodes of the graph are images and $W \in \mathbb{R}^{N\times N}$ is the graph adjacency matrix with $w_{ij} \in W$ measuring the strength of the edge linking x_i and x_j. The problem to be solved now is to discover a neighborhood set with high confidences for a given node $x_i \in X$. The confidence scores are usually recorded in an indicator vector $Y_i = [y_{i1}, y_{i2}, \dots, y_{iN}] \in \mathbb{R}^{1\times N}$.

DN borrows the idea of dominant set proposed in [28] and deems that the dominant neighbors of a given image, as a subset of its kNN, should correspond to a maximal clique in the affinity graph. It is formally defined as

$$\max_{Y_i} Y_i^{\mathrm{T}} W Y_i, s.t.\ Y_i \geq 0,\ \ Y_i \mathbf{1}^{\mathrm{T}} = 1, \tag{1}$$

where $\mathbf{1} \in \mathbb{R}^{1\times N}$ is a row vector with all elements equal to 1. The indicator function Y_i is subsequently learned by the replicator equation [29], as

$$y_{i,j}^{(t+1)} = y_{i,j}^{(t)} \frac{\left(WY_i^{(t)}\right)_j}{{Y_i^{(t)}}^{\mathrm{T}} W Y_i^{(t)}},\ \ j = 1, 2, \dots, N, \tag{2}$$

where t denotes the number of iterations. As demonstrated in [29], the update scheme presented in Eq. (2) is guaranteed to converge to a local maximizer after sufficient iterations.

Although DN achieves some improvements in retrieval performances, it still has some severe disadvantages. For example, it is prone to getting stuck at wrong local optima. In [24], a thorough analysis on DN is given and a new simple yet effective algorithm called CN is proposed. CN keeps track of the times that an image pair appears together among all rounds of kNN. The principle of CN is that if two images are similar, they tend to appear in the kNN of other images much more frequently. Instead of explicitly learning the indicator function Y, CN defines a consensus matrix $C \in \mathbb{R}^{N\times N}$. The update scheme of C is shown in Algorithm 1. Afterwards, the indicator function Y can be obtained via row-normalizing C. It is testified that CN can achieve quite stable performances compared with DN, especially with larger neighborhood size. However, it still lacks a theoretical guarantee, making it difficult to generalize well in diverse data structures.

Algorithm 1: The pseudocode of consensus of kNN.

```
Input:
$W \in \mathbb{R}^{N\times N}$: the affinity matrix;
Output:
$C \in \mathbb{R}^{N\times N}$: the consensus matrix.
begin
    Initialize $C = 0$;
    for $i = 1 : N$ do
        Obtain $kNN(x_i)$ by applying k-nearest neighbor rule to $W$;
        for $p = 1 : N$ do
            for $q = p + 1 : N$ do
                if $x_p \in kNN(x_i)$~and~$x_q \in kNN(x_i)$ then
                    $C(p, q) = C(p, q) + 1$;
                    $C(q, p) = C(q, p) + 1$;
                end
            end
        end
    end
    return $C$
end
```

As a smooth operator to preserve the local structure of data manifold, graph Laplacian has been applied to various applications, such as feature coding [30], feature selection [31], image annotation [32], semi-supervised learning [33], etc. The proposed algorithm, called Smooth Neighborhood (SN), is essentially based on the usage of graph Laplacian. It interprets the procedure of neighbor selection in a probabilistic manner similar to DN.

3. Proposed method

As analyzed above, neither the simplest k-Nearest Neighbor (kNN) rule nor some more advanced algorithms (e.g., Dominant Neighborhood [12] and Consensus of kNN [24]) cannot satisfy the manifold assumption. To remedy this, we propose a robust algorithm to select neighbors in an unsupervised way, formally defined as

$$\min_{Y} \sum_{i<j}^{N} w_{ij} \|Y_i - Y_j\|^2 + \mu \sum_{i=1}^{N} \|Y_i - I_i\|^2, \tag{3}$$

where $Y_i = [y_{i1}, y_{i2}, \dots, y_{iN}] \in \mathbb{R}^{1\times N}$ is the indicator function of x_i that describes the probability distribution of its neighbors, that is, $y_{ij} \in [0,\ 1]$ measures the likelihood of x_j being the true neighbor of x_i. Y_i has exactly the same meaning as the indicator vector used in dominant neighborhood [12]. $I_i \in \mathbb{R}^{1\times N}$ is the ith row of an identity matrix I, indicating that x_i initializes itself as its nearest neighbor.

As can be seen from Eq. (3), SN holds two assumptions for the behavior of neighbor selection. The left term emphasizes that the selection of neighbors should be smooth along the underlying manifold structure, i.e., two nearby points (large w_{ij}) x_i and x_j should yield similar neighbors, that is, their indicator functions Y_i and Y_j should have a small distance. The right term emphasizes that no matter how we update the indicator Y_i for node x_i, it shall still enforce itself as its neighbor as much as possible. The trade-off between the two terms is balanced by the regularization parameter $\mu > 0$, which is determined empirically.

Suppose given $M \geq 2$ affinity graphs $\mathcal{G}^{(v)} = \left(X, W^{(v)}\right)_{v=1}^{M}$, we now begin to study how to select neighbors smoothly on multiple affinity graphs. To this end, we impose a weight learning paradigm into Eq. (3) to measure the importance of graphs, thus leading to our final objective function

$$\min_{\alpha, Y} \sum_{v=1}^{M} {\alpha^{(v)}}^{\gamma} \sum_{i<j}^{N} w_{ij}^{(v)} \|Y_i - Y_j\|^2 + \mu \sum_{i=1}^{N} \|Y_i - I_i\|^2, s.t.$$
$$\sum_{v=1}^{M} \alpha^{(v)} = 1, 0 \leq \alpha^{(v)} \leq 1, \tag{4}$$

where $\alpha = \{\alpha^{(1)}, \alpha^{(2)}, \dots, \alpha^{(M)}\}$ is the weight of M affinity graphs, and $\gamma > 1$ is a weight controller that adjusts the weight distribution across multiple affinity graphs.

Three noteworthy comments should be made here. First, weight learning procedure in Eq. (4) is implemented by adding ${\alpha^{(v)}}^{\gamma}$ to the objective function, instead of using $\alpha^{(v)}$ directly. The reason behind this choice is that if $\alpha^{(v)}$ is used, the optimal solution of α is $\alpha^{(v)} = 1$ for the affinity graph with minimum cost and $\alpha^{(v)} = 0$ for the other graphs, in other words, only the smoothest affinity graph is actually used. It is not a good behavior since the complementary nature among different affinity graphs is neglected. By using an additional exponential variable γ, the objective function is not linear with respect to α. Thus, one can easily adjust the weight distribution of these affinity graphs by varying γ.

Second, one may note that we do not set M indicator functions $Y_i^{(v)}$ associated with different affinity graphs $\mathcal{G}^{(v)}$. Instead, only a shared indicator function Y_i is utilized for a given node x_i. Such a setup has an inborn advantage that the consensus information among these affinity graphs can be easily exploited, while the trivial aggregation of M indicator functions can be avoided afterwards. In other words, only one indicator function Y_i can be directly attained to define the neighborhood structure of x_i, though multiple affinity graphs are used. Last, Eq. (4) is equivalent to Eq. (3) when $M = 1$. As a result, Eq. (4) can be directly applied with an arbitrary number of input affinity graphs without modifications.

3.1. Optimization

For the sake of notation convenience, the objective function in Eq. (4) can be re-written in matrix form as

$$\mathcal{J} = \sum_{v=1}^{M} \alpha^{(v)^{\gamma}} Tr\Big(Y^{\mathrm{T}} L^{(v)} Y\Big) + \mu \|Y - I\|_F^2, \tag{5}$$

where $Y = [Y_1^{\mathrm{T}}, Y_2^{\mathrm{T}}, \dots, Y_N^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^{N\times N}$, $L^{(v)} \in \mathbb{R}^{N\times N}$ is the vth graph Laplacian matrix defined as $L^{(v)} = D^{(v)} - W^{(v)}$, and $D^{(v)} \in \mathbb{R}^{N\times N}$ is a diagonal matrix whose value $d_{ii}^{(v)} = \sum_{j=1}^{N} w_{ij}^{(v)}$. The two operators $Tr(\cdot)$ and $\|\cdot\|_F$ calculate the trace and the Frobenius norm of the input matrix, respectively.

As can be seen from Eq. (5), there are two variables to optimize, i.e., the indicator function Y and the graph weight α. Hence, we decompose it into two sub-problems, then adopt an alternative way to solve the optimization problem iteratively.

3.1.1. Fix α, update Y

To get the optimal solution of this sub-problem, we compute the partial derivative of $\mathcal{J}$ with respect to Y as

$$\begin{cases} \dfrac{\partial \mathcal{J}}{\partial Y} = 2\displaystyle\sum_{v=1}^{M} \alpha^{(v)^{\gamma}} L^{(v)} Y + 2\mu(Y - I) & (6) \\ \dfrac{\partial^2 \mathcal{J}}{\partial Y Y^{\mathrm{T}}} = 2\displaystyle\sum_{v=1}^{M} \alpha^{(v)^{\gamma}} (L^{(v)} \otimes I) + 2\mu(I \otimes I), & (7) \end{cases}$$

where $\otimes$ denotes the Kronecker product.

Since graph Laplacian matrix is known to be positive semi-definite, we can easily derive that $\sum_{v=1}^{M} \alpha^{(v)^{\gamma}} (L^{(v)} \otimes I)$ is also positive semi-definite, considering that $\alpha^{(v)} > 0$. Hence, the spectral radius of the Hessian matrix in Eq. (7) is lower-bounded by 2μ, which suggests that it is positive definite as long as $\mu > 0$. Consequently, the object function is convex with respect to the variable Y.

Therefore, by setting the first derivative in Eq. (6) to zero, the closed-form solution of Y can be derived as

$$Y = \mu \left(\sum_{v=1}^{M} \alpha^{(v)^{\gamma}} L^{(v)} + \mu I \right)^{-1}. \tag{8}$$

Note that $\sum_{v=1}^{M} \alpha^{(v)^{\gamma}} L^{(v)} + \mu I$ is invertible as it is a positive definite matrix.

3.1.2. Fix Y, update α

In order to minimize Eq. (4) with respect to the graph weight α, we utilize Lagrange Multiplier method. Taking the constraint $\sum_{v=1}^{M} \alpha^{(v)} = 1$ into consideration, the Lagrange function of $\mathcal{J}$ is

$$L(\mathcal{J}, \lambda) = \sum_{v=1}^{M} \alpha^{(v)^{\gamma}} Tr\Big(Y^{\mathrm{T}} L^{(v)} Y\Big) + \mu \|Y - I\|_F^2 - \lambda \left(\sum_{v=1}^{M} \alpha^{(v)} - 1 \right). \tag{9}$$

It can be derived that the partial derivatives with respect to $\alpha^{(v)}$ and λ are

$$\begin{cases} \dfrac{\partial L(\mathcal{J}, \lambda)}{\partial \alpha^{(v)}} = \gamma \alpha^{(v)^{(\gamma-1)}} Tr\Big(Y^{\mathrm{T}} L^{(v)} Y\Big) - \lambda, & (10) \\ \frac{\partial^2 L(\mathcal{J}, \lambda)}{\partial \alpha^{(v)^2}} = \gamma(\gamma - 1) \alpha^{(v)^{(\gamma-2)}} Tr\Big(Y^{\mathrm{T}} L^{(v)} Y\Big), & (11) \\ \dfrac{\partial L(\mathcal{J}, \lambda)}{\partial \lambda} = -\sum_{v=1}^{M} \alpha^{(v)} + 1. & (12) \end{cases}$$

Note that in this sub-problem, $\|Y - I\|_F^2$ is a constant, which can be omitted directly. According to the definition in Eq. (3), $Tr(Y^{\mathrm{T}} L^{(v)} Y)$ is larger than 0. As $\gamma > 1$, we can find that the second derivative with respect to $\alpha^{(v)}$ in Eq. (11) is also larger than 0, which means that the object function is also convex with respect to $\alpha^{(v)}$.

Therefore, by setting the two derivatives in Eqs. (10) and (12) to zero simultaneously, the Lagrange multiplier λ is eliminated and the optimal solution of $\alpha^{(v)}$ is obtained finally as

$$\alpha^{(v)} = \frac{\Big(Tr\Big(Y^{\mathrm{T}} L^{(v)} Y\Big)\Big)^{\frac{1}{1-\gamma}}}{\sum_{v'=1}^{M} \Big(Tr\Big(Y^{\mathrm{T}} L^{(v')} Y\Big)\Big)^{\frac{1}{1-\gamma}}}. \tag{13}$$

For clarification, we summarize the whole procedure of optimization in Algorithm 2. After obtaining the indicator Y_i for the given node x_i, one can take the nodes with the top-K largest non-zero confidence scores to constitute k-smooth neighbor (kSN) by analogy of the standard kNN. Other variants, such as ϵ-smooth neighbor, reciprocal kSN, can be also defined in a similar manner.

Algorithm 2: The pseudocode of smooth neighborhood.

```
Input:
$W^{(v)} \in \mathbb{R}^{N\times N}$, $1 \le v \le M$: the affinity matrices;
Two hyperparameters: $\gamma$ and $\mu$.
Output:
$Y \in \mathbb{R}^{N\times N}$: the probability distribution of neighbors.
begin
    Initialize $\alpha^{(v)} = \frac{1}{M}$;
    repeat
        Update $Y$ using Eq. (8);
        Update the weight $\alpha$ using Eq (13)
    until convergence;
    return $Y$
end
```

3.2. Remarks

In this section, we give several supplementary remarks on the proposed smooth neighborhood.

3.2.1. Convergence

The convergence of the above optimization is guaranteed. Let $Y^{(t)}$ and $\alpha^{(t)}$ denote the value of Y and α in the tth iteration, respectively. Since we find the corresponding optimal solution for each sub-problem, the following inequality holds

$$J\Big(Y^{(t)}, \alpha^{(t)}\Big) \ge J\Big(Y^{(t+1)}, \alpha^{(t)}\Big) \ge J\Big((Y^{(t+1)}, \alpha^{(t+1)}\Big). \tag{14}$$

As a consequence, by solving two sub-problems alternatively, the objective value J keeps decreasing monotonically as t increases. Meanwhile, as J is lower bounded by zero, the convergence of the proposed algorithm can be verified.

3.2.2. Affinity initialization

To construct the affinity graph, we need to specify the similarity matrix W. The most common way is to use Gaussian Kernel as

$$w_{ij} = \exp\left(-\frac{\|x_i - x_j\|_2^2}{\sigma_{ij}^2} \right), \tag{15}$$

where σ_{ij} is the bandwidth parameter that controls the speed of similarity decay. In retrieval task, it is crucial to select a good σ_{ij} for better performances. Using a proper σ_{ij} is expected to pull intra-class images together and push extra-class images apart. Numerous works have focused on this issue, and most previous affinity learning algorithms [9,25,27] use an adaptive Gaussian kernel. For example, it is defined in [21] as $\sigma_{ij} = \sigma_i \sigma_j$, where $\sigma_i = \|x_i - x_{K(i)}\|_2$ and $K(i)$ is the index of the Kth nearest neighbor of x_i.

Those adaptive kernels usually require additional parameters to fix empirically, making the entire framework sophisticated. In our approach, we set $\sigma_{ij} = \sigma$, a constant for all pairs of images following the recent survey paper [19] on affinity learning. It is more helpful to figure out which part really works using a constant for affinity initialization.

3.2.3. Hyperparameters

There are two hyperparameters involved in the proposed algorithm.

γ controls the weight distribution of affinity graphs. When $\gamma \to 1$, only the smoothest affinity graph is counted. When $\gamma \to \infty$, equal weights are achieved consequently. The determination of γ depends on the degree of complementary nature among these affinity graphs. Richer complementarity prefers a larger γ.

In the naive solution, where M affinity graphs are weighted combined, the search space to determine the optimal value of weights grows exponentially with respect to M. It is trivial to determine the weights in such an exhausted way. When $M \geq 2$, the time complexity becomes unacceptable. By contrast, we only use one parameter γ to model the graph weights, which significantly reduces the number of parameters in the proposed algorithm.

The other parameter μ actually reflects the degree of influence fastened by the node x_i itself. For example, if $\mu \to \infty$ (imitates the extremely large influence), the indicator Y_i degenerates into identity matrix I_i. It reveals that only x_i itself is selected as its neighbor finally and all the other nodes are discarded.

3.2.4. Complexity analysis

Since the proposed SN operates on the affinity graph, it generally requires $O(N^2)$ to construct the affinity graph as the relationship between each two data points will be estimated. Then, as shown in Fig. 1, SN needs to analyze the relationship between three data points. Hence, its time complexity is $O(N^3)$. From a perspective of mathematical optimization, Eq. (8) shows that SN needs to inverse a matrix of size $N \times N$. As is known, it usually needs $O(N^3)$ to compute the matrix inversion. In summary, the overall time complexity of SN is bounded by $O(N^3)$. It should be mentioned that it can be potentially accelerated using certain linear algebra techniques like the Coppersmith–Winograd algorithm.

Among the compared methods, k-Nearest Neighbor (kNN) is the most efficient one. Since it only needs the pairwise similarities between each two points, its time complexity is merely $O(N^2)$. As Eq. (2) suggests, the optimization of Dominant Neighbors (DN) [12] requires $O(N^3)$ to determine the neighborhood structures for all the data points in a certain database. Meanwhile, Algorithm 1 shows that the time complexity of Consensus of kNN (CN) [24] is also $O(N^3)$ due to the usage of three iterations on the entire database.

The comparison of running time will be given in the experiments.

4. Experiments

In this section, we will testify the validity of Smooth Neighborhood (SN) against other related algorithms, including k-Nearest Neighbor (kNN), Dominant Neighbors (DN) [12] and Consensus of kNN (CN) [24] on several visual retrieval tasks. In particular, we feed the neighborhood structure learned by SN into a representative context-sensitive similarity called Sparse Contextual Activation (SCA) [15]. The experimental results suggest that SN can yield state-of-the-art performances with shape retrieval, image retrieval and 3D model retrieval.

4.1. Shape retrieval

Shape retrieval and matching [34–36] has been a fundamental yet hot topic for a long time. Following [12,19,24], the effectiveness of smooth neighborhood is first evaluated with shape retrieval on the MPEG-7 dataset [37]. It consists of 1400 silhouette images divided into 70 categories, where each category has 20 shapes. Each shape serves as the query data in turn, and the number of correct returned shapes in the top-40 is counted. The retrieval performance is measured by the bull's eye score, i.e., the ratio of the number of correct hits to the largest possible hits (20×1400). Therefore, bull's eye score ranges from 0 to 100% and a larger score indicates a better retrieval performance.

On this dataset, we implement four different shape similarity measures that are extensively used to learn the shape manifold in related literature, including Inner Distance Shape Context (IDSC) [38], Shape Context (SC) [39], Aspect Shape Context (ASC) [40] and Articulation-invariant Representation (AIR) [41]. The baseline performances of the four similarities are 85.40, 86.79, 88.39 and 93.54, respectively.

4.1.1. Qualitative evaluation

Inspired by context-based re-ranking algorithms that leverage neighborhood set comparison directly for re-ranking (see [15,17,23]), we first adopt a simple and basic pipeline for the performance evaluation. Let $\mathcal{N}(x_q)$ and $\mathcal{N}(x_p)$ denote the neighborhood set of the query x_q and the database image x_p, respectively, obtained by a certain neighborhood analysis algorithm. A more faithful context-sensitive similarity can be defined via using the Jaccard similarity between two sets as

$$S(x_q, x_p) = \frac{|\mathcal{N}(x_q) \cap \mathcal{N}(x_p)|}{|\mathcal{N}(x_q) \cup \mathcal{N}(x_p)|}, \tag{16}$$

where $|\cdot|$ measures the cardinality of the input set. The motivation of Eq. (16) is straightforward, i.e., if two images are similar, they tend to have extensive common neighbors.

In Fig. 2, we plot the retrieval performances of Eq. (16) embedded with different neighborhood analysis algorithms as a function of neighborhood size. As analyzed above, almost all the previous works cannot deal with more than one affinity graph. In order to provide a fair comparison in this situation, the results of kNN, DN and CN are implemented using a linear combination of those graphs with equal weights. The parameter setup of the proposed SN is as follows. The weight controller $\gamma = 3$, the regularizer $\mu = 0.08$. For affinity initialization, we set $\sigma = 0.2$.

A first glance at Fig. 2 shows that the neighborhood structure generated by the proposed SN is much more robust than the other compared algorithms. Especially at larger K, the advantage of SN is more dramatic. It clearly demonstrates the benefit of exploring the local consistency in the proposed method. Since there are 20 shapes per category on the MPEG-7 dataset, outliers are likely to be included when the neighborhood size is larger than 20. Nevertheless, the objective function in Eq. (4) regularizes that even if a relatively larger K is specified, the behavior of selecting neighbors of nearby points is forced to be as similar as possible. Thus, we can find that the performance of SN is quite stable at variable neighborhood sizes. Such a nice property makes SN especially suitable to context-based re-ranking algorithms, where contextual information is described by neighborhood structures.

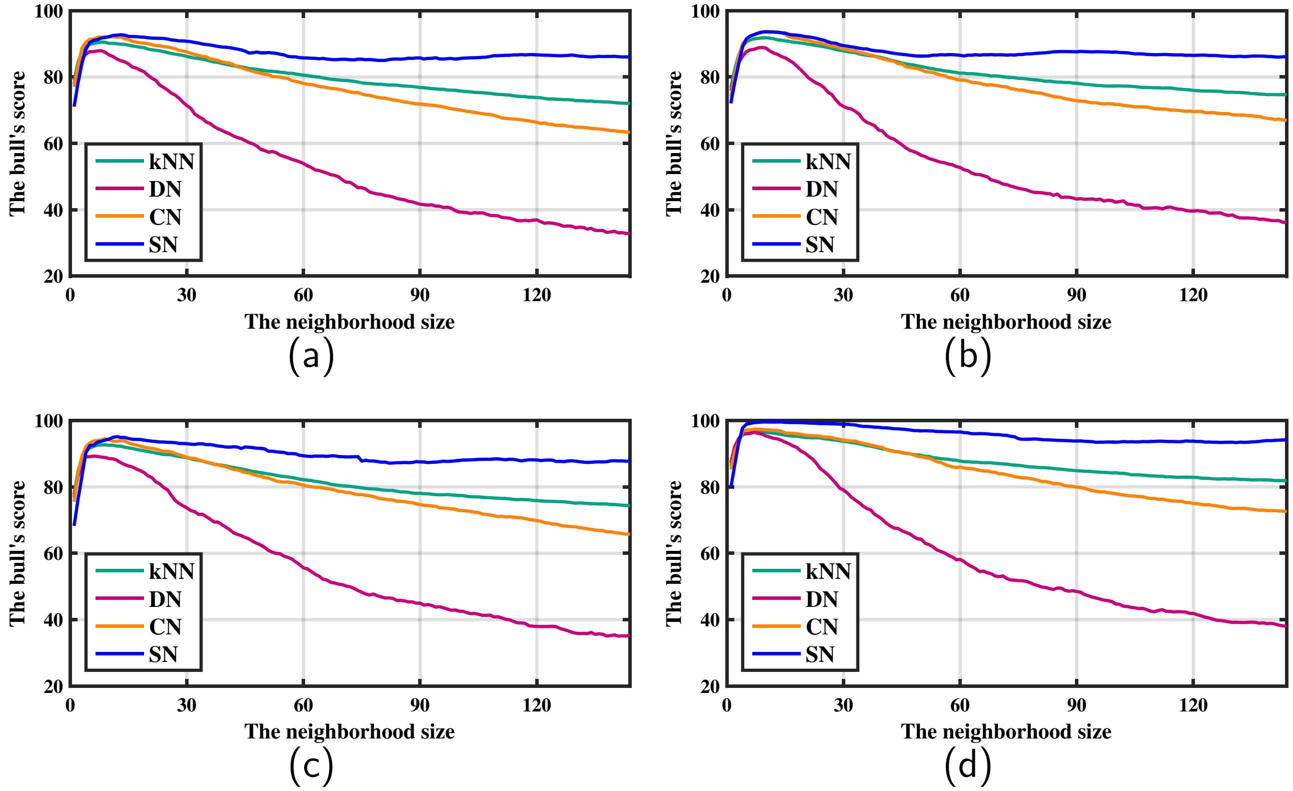

Fig. 2. The comparison using Eq. (16) on the MPEG-7 dataset. The baseline similarities used are IDSC (a), SC (b), ASC (c) and IDSC+SC+ASC (d), respectively.

In [24], CN is also demonstrated to provide stable performances at a relatively larger neighborhood size. However, it is simply inspired by an experimental observation, but lacks theoretical analysis. In comparison, the proposed SN provides an explicit objective function based on graph Laplacian to preserve the local manifold structure, so that the neighbor selection can be as smooth as possible along the underlying manifold. It can be also found that DN yields poor performances for two reasons. First, it starts to converge into false neighbors at larger K as claimed in [24]. Second, we do not give a good enough initialization for similarities using adaptive Gaussian kernels as previous work [12] do.

When fusing multiple affinity graphs, SN outperforms the other algorithms by a larger margin. The reason behind the superiority of SN is the weight learning mechanism imposed on multiple affinity graphs. On one hand, this learning paradigm can give prominence to the smoothest affinity graph and suppress the negative impacts from non-smooth affinity graphs. On the other hand, it can well exploit the complementary nature and consensus information among them, which is controlled by the weight controller γ. As introduced above, rich complementary nature prefers larger γ.

4.1.2. Improving context-sensitive similarity

There are many candidate context-sensitive similarities to distinguish the discriminative power of neighborhood analysis algorithms (see the framework of diffusion process in [19] for a comprehensive summary). However, the convergence of most these algorithms is not guaranteed, so the iteration has to be stopped at a "right" moment. Whereas, it is generally intractable in retrieval to set the stop condition without annotated training data. Moreover, our experimental results show that some methods heavily rely on a proper initialization for the pairwise similarities using adaptive Gaussian kernel as introduced in Section 3.2, i.e., σ_{ij} should be different for each pair of data points. If σ_{ij} is set to a constant as in this paper, these iterative algorithms, including Locally Constrained Diffusion Process (LCDP) [25], are prone to producing incorrect dense clusters. Such an experimental observation (refer to Table 2) usually occurs when the performances of the input baselines are relatively low (e.g., IDSC). Finally, diffusion process usually consists of three parts: affinity initialization, the definition of transition matrix and the definition of the update scheme. Nevertheless, the neighborhood analysis techniques only pay attention to the definition of transition matrix, since it records the neighborhood structures to constrain the diffusion process locally. In this sense, these variants of diffusion process are not proper to assess the neighborhood analysis techniques.

Instead of using LCDP, we turn to a recent re-ranking algorithm called Sparse Contextual Activation (SCA) [15], which is simple to implement and insensitive to parameter tuning. SCA is particularly suitable to evaluate the neighborhood analysis techniques since it directly focuses on the usage of the neighborhood structure by comparing two neighborhood sets in fuzzy set theory. It has two parameters to be fixed manually, i.e., the parameters k_1 and k_2 determining the first-order and the second-order neighborhood sizes, respectively. In our case, the two neighborhood sets can be obtained using kNN, DN, CN and the proposed SN.

Table 1 presents the performance comparison, and the results of kNN, DN and CN are reported at its optimal parameter setup (not necessarily the same parameter setup). First, we could observe that in line with previous analysis, the proposed SN achieves the best performances among the compared algorithms no matter what baseline similarities are used. Second, when the baseline similarity varies, the second best performances are achieved by different algorithms, and no algorithms generalize well in diverse data structures parameterized by different baseline similarities. For example, excluding SN, the simplest kNN rule outperforms others when AIR serves as the baseline similarity. Nevertheless, it can be clearly drawn that SN possesses consistent performance gain over the second best. Third, the performance gain of

Table 1
The performance comparison in bull's eye score (%) of neighborhood analysis algorithms on the MPEG-7 dataset. The best and the second best results are marked with red and green color, respectively. The last column presents the performance gain of SN over the best-performing algorithms.

Baselines	kNN	DN	CN	SN	Gain
IDSC	90.07	90.07	92.70	93.52	+0.82
SC	93.23	89.85	93.97	95.25	+1.28
ASC	94.34	92.49	95.25	95.98	+0.73
AIR	99.97	98.84	99.85	100.00	+0.03
IDSC+SC	98.20	96.96	97.61	99.25	+1.05
IDSC+SC+ASC	97.61	98.25	97.86	99.81	+1.56

SN is more dramatic when lower baselines are used. For instance, the performance gain over the second best is only 0.03 with the relatively high baseline AIR. When the relatively low baseline like SC is used, the performance gain increases to 1.28. When multiple similarity measures (e.g., IDSC + SC or IDSC + SC + ASC) are integrated, the superiorly of SN is more distinctive. In the following experiments, we will combine SCA with SN to provide the retrieval performances of our method, if not specified otherwise.

4.1.3. Comparison with state-of-the-art

In Table 2, we give a thorough comparison with the state-of-the-art algorithms. The results in the table are carefully classified according to the type of input baseline similarity. Since Generic Diffusion Process (GDP) [19] only reports its result with AIR as the input similarity, its performances with IDSC, SC and ASC are implemented by the authors using the public available codes,[1] thus marked with ⋆ on the upper right corner.

IDSC is the most frequently used baseline shape similarity. Combining smooth neighborhood and SCA, we report a new level performance using IDSC as the baseline similarity, which is **93.52** in bull's eye score. Of course, it is not the best performance on this dataset, since some context-based re-ranking algorithms take the higher baseline AIR as the input similarity. For instance, Tensor Product Graph (TPG) [12] reports 99.99 bull's eye score by combining adaptive Gaussian kernel, dominant neighbor and diffusion process. By contrast, we achieve the perfect score **100** by simply using smooth neighborhood and SCA. The performance gain is especially valuable when considering the fact that we do not use a more accurate similarity initialization using adaptive Gaussian kernel as TPG. Meanwhile, generic diffusion process [19] also reports 100 bull's eye score by enumerating 72 variants of diffusion process (4 different affinity initializations, 6 different transition matrices and 3 different update schemes). One may note that the retrieval performances of generic diffusion process are inferior with IDSC, SC or ASC. It verifies our previous claim that a proper affinity initialization using adaptive Gaussian kernel is crucial for diffusion process when lower baseline similarities are used. A recent algorithm [42] also correctly reveal this intrinsic data structure by exploiting the Reciprocal kNN Graph and its Connected Components.

Previous similarity fusion algorithms usually only consider integrating two similarity measures, e.g., SC and IDSC. This is due to the fact that most of them [11] are based on the co-training framework that is only suitable to deal with two similarity measures. Our method potentially provides an alternative way of similarity fusion at the neighbor selection level. What is more important is that it is not limited to two similarity measures. Even though using more similarity measures, we can still obtain only one shared neighborhood structure, as well as the weights of different affinity graphs. It can be expected that when more complementary similarities are fused, a more robust neighborhood structure can be learned thus leading to higher retrieval performances. To support our speculation, we also report the performances of SN with a combination of similarities that is not used by previous works. For example, by combining IDSC, SC and ASC, SN can yield bull's eye score **99.81**. To our best knowledge now, it is the best performance on the MPEG-7 dataset while AIR is not used.

Table 2
The bull's eye scores (%) of different methods on the MPEG-7 dataset.

Descriptors	Methods	Bull's eye score
IDSC	Contextual Dissimilarity Measure (CDM) [8]	88.30
IDSC	Generic Diffusion Process (GDP)⋆ [19]	90.96
IDSC	Index-Based Re-Ranking [14]	91.56
IDSC	Graph Transduction (GT) [9]	91.61
IDSC	Locally Constrained Diffusion Process [25]	92.36
IDSC	RL-Sim Re-Ranking [2]	92.62
IDSC	Shortest Path Propagation (SSP) [10]	93.35
IDSC	Mutual kNN Graph (mkNN) [21]	93.40
IDSC	Sparse Contextual Activation (SCA) [15]	93.44
IDSC	**Smooth Neighborhood (Ours)**	**93.52**
SC	Generic Diffusion Process (GDP)⋆ [19]	92.81
SC	Graph Transduction (GT) [9]	92.91
SC	Sparse Contextual Activation (SCA) [15]	95.21
SC	**Smooth Neighborhood (Ours)**	**95.25**
ASC	Generic Diffusion Process (GDP)⋆ [19]	93.95
ASC	Index-Based Re-Ranking [14]	94.09
ASC	RL-Sim Re-Ranking [2]	95.75
ASC	Locally Constrained DP (LCDP) [25]	95.96
ASC	Tensor Product Graph (TPG) [12]	**96.47**
ASC	**Smooth Neighborhood (Ours)**	95.98
IDSC+SC	Co-Transduction [11]	97.72
IDSC+SC	Locally Constrained Mixed Diffusion (LCMD) [27]	98.84
IDSC+SC	Sparse Contextual Activation (SCA) [15]	99.01
IDSC+SC	**Smooth Neighborhood (Ours)**	**99.25**
AIR	Tensor Product Graph (TPG) [12]	99.99
AIR	Generic Diffusion Process (GDP) [19]	100.00
AIR	Connected Components [42]	100.00
AIR	**Smooth Neighborhood (Ours)**	**100.00**

[1] Available at http://vh.icg.tugraz.at/index.php?content=topics/diffusion.php

Table 3
The N-S scores of different methods on the Ukbench dataset. Note that Query Adaptive Fusion uses 5 input similarities, and the last result of our method is produced by using all the 4 similarities implemented in this paper.

Descriptors	Methods	N-S score
BoW (3.52)	kNN Re-ranking [47]	3.56
BoW (3.22)	Tensor Product Graph [12]	3.61
BoW (3.26)	Co-transduction [11]	3.66
BoW (3.50)	RNN Re-ranking [22]	3.67
BoW (3.54)	Graph Fusion [17]	3.67
BoW (3.33)	Contextual Dissimilarity Measure [8]	3.68
BoW (3.56)	Sparse Contextual Activation [15]	3.69
BoW (3.57)	**Smooth Neighborhood (Ours)**	**3.75**
CNN (3.44)	**Smooth Neighborhood (Ours)**	**3.66**
CNN (3.65)	**Smooth Neighborhood (Ours)**	**3.81**
HSV (3.17)	Graph Fusion [17]	3.28
HSV (3.40)	Sparse Contextual Activation [15]	3.56
HSV (3.40)	**Smooth Neighborhood (Ours)**	**3.56**
BoW (3.20,3.17,2.81)	Locally Constrained Mixed Diffusion [27]	3.70
BoW (3.54), HSV (3.17)	Graph Fusion [17]	3.77
BoW (3.54), HSV (3.17)	Graph Fusion [18]	3.83
BoW (3.58), CNN (3.40), etc.	Query Adaptive Fusion [3]	3.84
BoW (3.56), HSV (3.40)	Sparse Contextual Activation [15]	3.86
BoW (3.13), CNN (3.87)	ONE [48]	3.89
BoW (3.54), CNN (3.31)	Connected Components [42]	3.89
BoW (3.57), CNN (3.44), etc.	**Smooth Neighborhood (Ours)**	**3.98**

4.2. Image retrieval

The proposed approach is then evaluated on the Ukbench dataset [43], commonly used as a benchmark in natural image retrieval. It is comprised of 2550 objects and each object has 4 different viewpoints or illuminations. All 10,200 images are both indexed as queries and database images. The most widely-used evaluation metric is N-S score, which counts the average recall of the top-4 ranked images. Hence, N-S score ranges from 1 implying only the query itself is returned, to 4 that is the perfect N-S score on this dataset.

In this experiment, we implement 4 kinds of similarity measures. They are

1. Bag of Words (BoW): SIFT [44] descriptors are extracted at interest points produced by Hessian-affine detectors, and later converted to RootSIFT [45]. A codebook with 20k entries is learned with K-means on independent data. We follow the pipeline of Hamming embedding [46] that uses cosine similarity for affinity initialization. The N-S score of BoW representation is 3.57.
2. Convolutional Neutral Network (CNN): Two CNN features are extracted based on the trained AlexNet model. The activations of 5th convolutional layer and the 7th fully-connected layer are used. For each image, the activation is first square-rooted then L_2 normalized. The N-S scores of the two CNN features are 3.44 and 3.65, respectively.
3. HSV: Following [17], we extract 1000 dimensional HSV color histogram ($20 \times 10 \times 5$ bins for H, S, V components, respectively). The HSV histogram is first L_1 normalized then square-rooted. The N-S score of HSV is 3.40.

The parameter setup of SN is the same as those reported in Section 4.1. $\sigma = 0.5$ is used for affinity initialization.

Extensive algorithms have reported their performances on the Ukbench dataset. However, in Table 3, we only collect two kinds of results for comparison, i.e., postprocessing algorithms such as context-sensitive similarities, and the state-of-the-art performances ever reported. To improve the readability, the results are ordered from those using single feature to those using multiple features. Meanwhile, since the performances of baselines used by different methods are usually quite different in natural image retrieval, we also include N-S scores of those baselines in the parentheses.

Graph Fusion [17] is a representative algorithm that integrates multiple affinity graphs by averaging the strength of edges with equal weights. It reports 3.77 N-S score by fusing local SIFT feature and holistic HSV color histogram, and later reports 3.83 in [18] by iteratively constructing the graph. In a sense, the proposed SN can be also considered as a kind of graph fusion. However, the difference is that we do not consider simply averaging the edge weights. Instead, SN tries to find a robust neighborhood structure shared by different affinity graphs, so that the consensus information among them can be largely preserved. In [3], N-S score 3.84 is achieved by fusing five kinds of features. Using BoW and HSV as the input similarities, SCA [15] reports N-S score 3.86. Fusing BoW and CNN feature, ONE [48] and Connected Components [42] both achieve N-S score 3.89, which is the best performance to our knowledge.

Besides those postprocessing methods, [49] reports NS score 3.67 by using query expansion. Sun et al. [50] extracts object-level features rather than traditional local or global features, and reports 3.81. Paulin et al. [51] and Babenko and Lempitsky [52] exploit local convolutional features, and report N-S score 3.76 and 3.65, respectively. In this paper, we achieve the near perfect N-S score **3.98** by fusing all the 4 similarities, including BoW, two CNN features and HSV. It outperforms the previous state-of-the-art remarkably.

4.3. 3D model retrieval

3D model [53,54] is a more complicated data pattern, which is of great academic value. In this section, we test the performance of SN on two widely-accepted 3D model retrieval datasets, i.e., Princeton Shape Benchmark (PSB) [55] and Watertight Models track of SHape REtrieval Contest 2007 dataset (WM-SHREC07) [56].

PSB dataset is a classic benchmark for generic 3D model retrieval, which is comprised of 1804 3D polygonal models. The entire dataset is divided into training set and testing set with 907 models each. Following the common settings, only the testing set with 92 categories is used to evaluate the performance of unsupervised retrieval. SHape REtrieval Contest (SHREC), held each year, is an authoritative competition for evaluating the effectiveness of 3D object retrieval algorithms. In this paper, we use WM-SHREC07,

Table 4
The baseline performances (%) on the PSB and the WM-SHREC07 datasets.

Methods	PSB				WM-SHREC07			
	NN	FT	ST	DCG	NN	FT	ST	DCG
VLAD [57]	80.4	57.5	71.5	79.9	95.7	70.8	80.9	90.3
NSC [58]	79.4	57.2	69.8	78.5	96.5	78.3	90.5	93.1

which contains 400 watertight mesh models that are evenly distributed into 20 classes.

To quantify the performance, we employ the following evaluation metrics:

- Nearest Neighbor (NN): the percentage of the closest matches that belongs to the same class as the query.
- First Tier (FT): the recall for the top $(C-1)$ matches in the ranked list, where C is the number of shapes in the category that query belongs to.
- Second Tier (ST): the recall for the top $2(C-1)$ matches in the ranked list, where C is the number of shapes in the category that query belongs to.
- Discounted Cumulative Gain (DCG): a statistic that attaches more importance to the correct results near the front of the ranked list than the correct results at the end of the ranked list, under the assumption that a user is more likely to consider the retrieved candidates in the front of the list.

All the evaluation metrics range from 0 to 100%, and higher values indicate better performances. One can refer to [55] for their formal definitions.

In this experiment, the same two baseline similarities used by SCA [15] are tested, as

1. VLAD: SIFT [44] descriptors are extracted at the 64 depth projections of the 3D models. Then, Vector of Locally Aggregated Descriptors (VLAD) [57] is used to encode those descriptors with a codebook of 2048 entries. The generated features are L_2 normalized, and the pairwise shape similarity is computed in the Euclidean metric.
2. NSC [58]: Neural shape codes (NSC) trains a Convolutional Neural Network on the projections of the 3D model. Then, it extracts the activations of the internal layers as the feature representations of the projections. At last, Hausdorff distance is employed for shape matching.

The performances of those two baselines are given in Table 4. The parameter setup of SN is the same as those reported in Section 4.2.

The performance comparison with other representative methods is given in Table 5. kNN is used by SCA in [15] to define the contextual information of data points, while in this paper we replace it with the proposed SN. As the input baseline similarities are exactly the same, one can clearly observe the superiority of SN over kNN. As presented in Table 5, SCA originally achieves ST 81.0 on the PSB dataset and 95.6 on the WM-SHREC07 dataset. After SN is used, SCA improves this score to 82.5 and 97.2, respectively. It demonstrates again that with a more faithful context provided by SN, SCA can better reveal the intrinsic relationship between data points. Moreover, SN also outperforms other representative methods on all the evaluation metrics remarkably. Also, it can be expected that if more robust descriptors [54,68,69] are used, the performance will be better.

4.4. Discussion

4.4.1. Execution time

Table 6 compares the execution time of the four neighborhood analysis algorithms with different baseline similarities on the MPEG-7 dataset. To make the comparison straightforward, we only count the time cost of neighbor selection, while that of other procedures like feature extraction and indexing are excluded. All the experiments are done on a personal computer with an Intel(R) Core(TM) i5 CPU (3.40 GHz) and 16 GB memory.

As analyzed in Section 3.2.4, kNN is one order of magnitude faster than the others. Hence, it can be easily found that kNN only needs around 0.2*s* in total to determine the neighborhood structure. In comparison, the execution time of DN, CN and the proposed SN is much longer. As DN has to traverse the entire database for 3 times, it is the most inefficient one. Owing to the well-defined matrix multiplication and inverse in common computing platforms, the time cost of CN and SN is relatively lower. In particular for the proposed SN, the time cost is much heavier when multiple similarities are fused, as SN needs to perform alternative optimization in this case. Note that the cost of SN can be further reduced if more advanced optimization techniques are used (e.g., parallel computing). Nevertheless, it is still an open issue to design more efficient neighbor selection algorithms.

4.4.2. Parameter sensitivity

In the experiments above, the values of the hyperparameters γ and μ are fixed empirically. In Fig. 3, we depict their influences on the retrieval accuracy. The discussion is done on the MPEG-7 dataset with SC and IDSC as the baseline similarities. As can be found, the proposed SN is insensitive to the change of the hyperparameters. It is also observed that if carefully tuning the parameters, SN can achieve a better performance. For example, Fig. 3(a) shows that the best bull's eye score is 99.52, which is higher than 99.25 reported in Table 2.

5. Conclusions

In this paper, we propose a neighbor selection algorithm called Smooth Neighborhood (SN). Compared with related algorithms, the two key advantages of SN are the theoretical guarantee of the underlying manifold structure and the capacity of dealing with multiple affinity graphs. Embedded with context-sensitive similarities, SN is evaluated on retrieval tasks and achieves much better performances than other related algorithms, including kNN, dominant neighborhood and consensus of kNN. In particular, despite the perfect bull's eye score with shape retrieval on the MPEG-7 dataset, SN also achieves the near perfect N-S score 3.98 with natural image retrieval on the Ukbench dataset, and the state-of-the-art performances with 3D model retrieval on the PSB and WM-SHREC07 datasets.

Since the proposed method focuses on neighborhood analysis, it can be potentially plunged into other retrieval systems (e.g., RNN Re-ranking [22], kNN Re-ranking [47]) and other computer vision tasks (e.g., graph matching [70], image categorization, object detection, video search [71]), where a more robust neighborhood structure is required. Moreover, it should be addressed that the weight learning paradigm in our method is exerted into the entire affinity graphs. However, it is known that query specific weight is a more proper choice in retrieval task. We would like to exploit these issues in the future.

Acknowledgment

This work was supported by National Key R&D Program of China (No. 2018YFB1004600), NSFC 61573160 and 61429201. This work was supported to Dr. Xiang Bai by the National Program for Support of Top-notch Young Professionals and the Program for HUST Academic Frontier Youth Team, to Dr. Qi Tian by ARO grants W911NF-15-1-0290 and Faculty Research Gift Awards by NEC Laboratories of America and Blippar. We thank Jun Guo from Tsinghua-

Table 5
The performance comparison (%) with other state-of-the-art algorithms on the PSB and the WM-SHREC07 datasets.

Methods	PSB				WM-SHREC07			
	NN	FT	ST	DCG	NN	FT	ST	DCG
LFD [59]	65.7	38.0	48.7	64.3	92.3	52.6	66.2	–
Tabia et al.[60]	–	–	–	–	85.3	52.7	63.9	71.9
DESIRE [61]	66.5	40.3	51.2	66.3	91.7	53.5	67.3	–
tBD [62]	72.3	–	–	66.7	–	–	–	–
Covariance [63]	–	–	–	–	93.0	62.3	73.7	86.4
Spatially-covariance [64]	–	–	–	–	92.5	65.4	75.6	86.2
2D/3D Hybrid [65]	74.2	47.3	60.6	–	95.5	64.2	77.3	–
PANORAMA [66]	75.2	53.1	65.9	–	95.7	74.3	83.9	–
3DVFF [67]	–	–	–	84.1	–	–	–	–
SCA [15]	**83.7**	70.0	81.0	85.0	99.0	90.0	95.6	97.2
Smooth Neighborhood	81.0	**70.8**	**82.5**	**85.2**	**99.3**	**92.4**	**97.2**	**97.8**

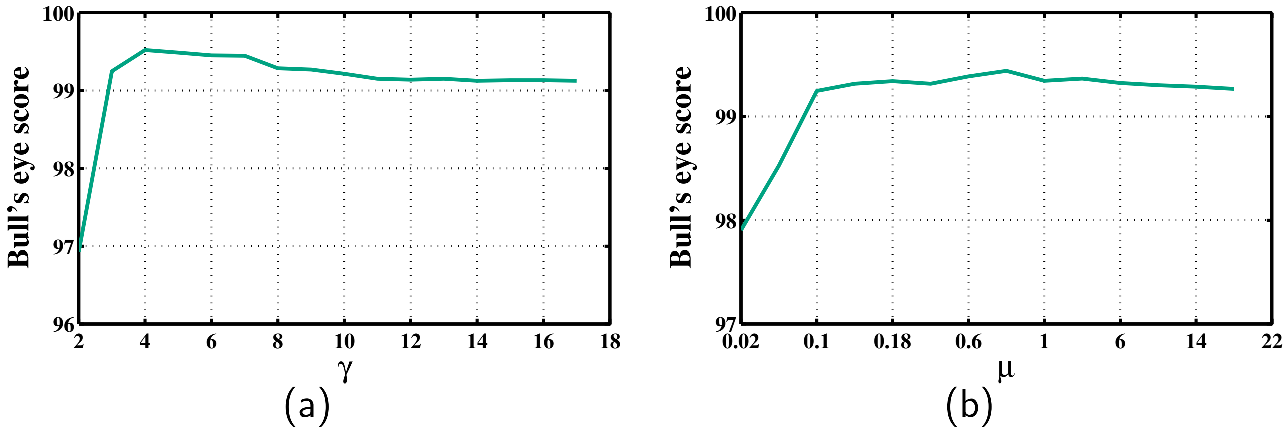

Fig. 3. The influence of γ (a) and μ (b) on the retrieval accuracy.

Table 6
The comparison in execution time (s) of neighborhood analysis algorithms on the MPEG-7 dataset.

Baselines	kNN	DN [12]	CN [24]	SN (Ours)
IDSC	**0.20**	1418	2.81	4.14
SC+IDSC	**0.21**	1419	3.21	29.40

Berkeley Shenzhen Institute for the helpful and instructive discussions.

References

[1] M.S. Lew, N. Sebe, C. Djeraba, R. Jain, Content-based multimedia information retrieval: state of the art and challenges, ACM Trans. Multimed. Comput. Commun. Appl. 2 (1) (2006) 1–19.

[2] D. Pedronette, R. Torres, Image re-ranking and rank aggregation based on similarity of ranked lists, Pattern Recognit. 46 (8) (2013) 2350–2360.

[3] L. Zheng, S. Wang, L. Tian, F. He, Z. Liu, Q. Tian, Query-adaptive late fusion for image search and person re-identification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1741–1750.

[4] Y. Yang, F. Nie, D. Xu, J. Luo, Y. Zhuang, Y. Pan, A multimedia retrieval framework based on semi-supervised ranking and relevance feedback, IEEE Trans. Pattern Anal. Mach. Intell. 34 (4) (2012) 723–742.

[5] Y. Yang, J. Song, Z. Huang, Z. Ma, N. Sebe, A.G. Hauptmann, Multi-feature fusion via hierarchical regression for multimedia analysis, IEEE Trans. Multimed. 15 (3) (2013) 572–581.

[6] T. Mei, Y. Rui, S. Li, Q. Tian, Multimedia search reranking: a literature survey, ACM Comput. Surv. 46 (3) (2014) 38.

[7] Z.-J. Zha, L. Yang, T. Mei, M. Wang, Z. Wang, T.-S. Chua, X.-S. Hua, Visual query suggestion: towards capturing user intent in internet image search, ACM Trans. Multimed. Comput. Commun. Appl. 6 (3) (2010) 13.

[8] H. Jegou, C. Schmid, H. Harzallah, J.J. Verbeek, Accurate image search using the contextual dissimilarity measure, IEEE Trans. Pattern Anal. Mach. Intell. 32 (1) (2010) 2–11.

[9] X. Bai, X. Yang, L.J. Latecki, W. Liu, Z. Tu, Learning context-sensitive shape similarity by graph transduction, IEEE Trans. Pattern Anal. Mach. Intell. 32 (5) (2010) 861–874.

[10] 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.

[11] X. Bai, B. Wang, X. Wang, W. Liu, Z. Tu, Co-transduction for shape retrieval, in: Proceedings of the European Conference on Computer Vision, 2010, pp. 328–341.

[12] X. Yang, L.J. Latecki, Affinity learning on a tensor product graph with applications to shape and image retrieval, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2011, pp. 2369–2376.

[13] B. Wang, Z. Tu, Affinity learning via self-diffusion for image segmentation and clustering, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 2312–2319.

[14] D. Pedronette, J. Almeida, R. Torres, A scalable re-ranking method for content-based image retrieval, Inf. Sci. (Ny) 265 (2014) 91–104.

[15] S. Bai, X. Bai, Sparse contextual activation for efficient visual re-ranking, IEEE Trans. Image Process. 25 (3) (2016) 1056–1069.

[16] Y. Chen, X. Li, A. Dick, R. Hill, Ranking consistency for image matching and object retrieval, Pattern Recognit. 47 (3) (2014) 1349–1360.

[17] S. Zhang, M. Yang, T. Cour, K. Yu, D.N. Metaxas, Query specific fusion for image retrieval, in: Proceedings of the European Conference on Computer Vision, 2012, pp. 660–673.

[18] S. Zhang, M. Yang, T. Cour, K. Yu, D.N. Metaxas, Query specific rank fusion for image retrieval, IEEE Trans. Pattern Anal. Mach. Intell. 37 (4) (2015) 803–815.

[19] M. Donoser, H. Bischof, Diffusion processes for retrieval revisited, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 1320–1327.

[20] S.T. Roweis, L.K. Saul, Nonlinear dimensionality reduction by locally linear embedding, Science 290 (5500) (2000) 2323–2326.

[21] P. Kontschieder, M. Donoser, H. Bischof, Beyond pairwise shape similarity analysis, in: Proceedings of the Asian Conference on Computer Vision, 2009, pp. 655–666.

[22] 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 IEEE Conference on Computer Vision and Pattern Recognition, 2011, pp. 777–784.

[23] D. Pedronette, O. Penatti, R. Torres, Unsupervised manifold learning using reciprocal kNN graphs in image re-ranking and rank aggregation tasks, Image Vis. Comput. 32 (2) (2014) 120–130.

[24] V. Premachandran, R. Kakarala, Consensus of kNNs for robust neighborhood selection on graph-based manifolds, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 1594–1601.
[25] 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 IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 357–364.
[26] S. Bai, S. Sun, X. Bai, Z. Zhang, Q. Tian, Smooth neighborhood structure mining on multiple affinity graphs with applications to context-sensitive similarity, in: Proceedings of the European Conference on Computer Vision, Springer, 2016, pp. 592–608.
[27] L. Luo, C. Shen, C. Zhang, A. van den Hengel, Shape similarity analysis by self–tuning locally constrained mixed-diffusion, IEEE Trans. Multimed. 15 (5) (2013) 1174–1183.
[28] M. Pavan, M. Pelillo, Dominant sets and pairwise clustering, IEEE Trans. Pattern Anal. Mach. Intell. 29 (1) (2007) 167–172.
[29] M. Pelillo, Matching free trees with replicator equations, Adv. Neural Inf. Process. Syst. (2002) 865–872.
[30] S. Gao, I.W.-H. Tsang, L.-T. Chia, P. Zhao, Local features are not lonely–Laplacian sparse coding for image classification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2010, pp. 3555–3561.
[31] Z. Zhang, L. Bai, Y. Liang, E. Hancock, Joint hypergraph learning and sparse regression for feature selection, Pattern Recognit 63 (2017) 291–309.
[32] J. Wang, S.-F. Chang, X. Zhou, S.T. Wong, Active microscopic cellular image annotation by superposable graph transduction with imbalanced labels, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2008, pp. 1–8.
[33] X. Zhu, Z. Ghahramani, J. Lafferty, et al., Semi-supervised learning using gaussian fields and harmonic functions, in: Proceedings of the International Conference on Machine Learning, 2003, pp. 912–919.
[34] O. Arandjelovic, Matching objects across the textured-smooth continuum, in: Proceedings of the Australasian Conference on Robotics and Automation, 2012, pp. 1–8.
[35] R.A. Guler, S. Tari, G. Unal, Landmarks inside the shape: shape matching using image descriptors, Pattern Recognit. 49 (2016) 79–88.
[36] O. Arandjelović, Discriminative extended canonical correlation analysis for pattern set matching, Mach. Learn. 94 (3) (2014) 353–370.
[37] L.J. Latecki, R. Lakämper, U. Eckhardt, Shape descriptors for non-rigid shapes with a single closed contour, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2000, pp. 424–429.
[38] H. Ling, D.W. Jacobs, Shape classification using the inner-distance, IEEE Trans. Pattern Anal. Mach. Intell. 29 (2) (2007) 286–299.
[39] S. Belongie, J. Malik, J. Puzicha, Shape matching and object recognition using shape contexts, IEEE Trans. Pattern Anal. Mach. Intell. 24 (4) (2002) 509–522.
[40] H. Ling, X. Yang, L.J. Latecki, Balancing deformability and discriminability for shape matching, in: Proceedings of the European Conference on Computer Vision, 2010, pp. 411–424.
[41] R. Gopalan, P. Turaga, R. Chellappa, Articulation-invariant representation of non-planar shapes, in: Proceedings of the European Conference on Computer Vision, 2010, pp. 286–299.
[42] D.C.G. Pedronette, F.M.F. Gonçalves, I.R. Guilherme, Unsupervised manifold learning through reciprocal kNN graph and connected components for image retrieval tasks, Pattern Recognit 75 (2018) 161–174.
[43] D. Nistér, H. Stewénius, Scalable recognition with a vocabulary tree, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2006, pp. 2161–2168.
[44] D.G. Lowe, Distinctive image features from scale-invariant keypoints, Int. J. Comput. Vis. 60 (2) (2004) 91–110.
[45] R. Arandjelović, A. Zisserman, Three things everyone should know to improve object retrieval, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 2911–2918.
[46] H. Jegou, M. Douze, C. Schmid, Hamming embedding and weak geometric consistency for large scale image search, in: Proceedings of the European Conference on Computer Vision, 2008, pp. 304–317.
[47] X. Shen, Z. Lin, J. Brandt, S. Avidan, Y. Wu, Object retrieval and localization with spatially-constrained similarity measure and kNN re-ranking, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 3013–3020.
[48] L. Xie, R. Hong, B. Zhang, Q. Tian, Image classification and retrieval are one, in: Proceedings of the International Conference on Multimedia Retrieval, 2015, pp. 3–10.
[49] G. Tolias, H. Jégou, Visual query expansion with or without geometry: refining local descriptors by feature aggregation, Pattern Recognit. 47 (10) (2014) 3466–3476.
[50] S. Sun, W. Zhou, Q. Tian, H. Li, Scalable object retrieval with compact image representation from generic object regions, ACM Trans. Multimed. Comput. Commun. Appl. 12 (2) (2016) 29.
[51] M. Paulin, M. Douze, Z. Harchaoui, J. Mairal, F. Perronin, C. Schmid, Local convolutional features with unsupervised training for image retrieval, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 91–99.
[52] A. Babenko, V. Lempitsky, Aggregating deep convolutional features for image retrieval, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1269–1277.
[53] S. Bai, X. Bai, Z. Zhou, Z. Zhang, L.J. Latecki, Gift: a real-time and scalable 3D shape search engine, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 5023–5032.
[54] C. Wang, M. Pelillo, K. Siddiqi, Dominant set clustering and pooling for multi-view 3D object recognition, in: Proceedings of the British Machine Vision Conference, 2017.
[55] P. Shilane, P. Min, M.M. Kazhdan, T.A. Funkhouser, The princeton shape benchmark, in: Proceedings of the Shape Modeling Applications, 2004.
[56] D. Giorgi, S. Biasotti, L. Paraboschi, Shape retrieval contest 2007: watertight models track, in: Proceedings of the SHREC Competition, 8, 2007.
[57] H. Jegou, F. Perronnin, M. Douze, J. Sánchez, P. Perez, C. Schmid, Aggregating local image descriptors into compact codes, IEEE Trans. Pattern Anal. Mach. Intell. 34 (9) (2012) 1704–1716.
[58] S. Bai, X. Bai, W. Liu, F. Roli, Neural shape codes for 3D model retrieval, Pattern Recognit. Lett. 65 (2015) 15–21.
[59] D.Y. Chen, X.P. Tian, Y.T. Shen, M. Ouhyoung, On visual similarity based 3D model retrieval, Comput. Gr. Forum. 22 (3) (2003) 223–232.
[60] H. Tabia, M. Daoudi, J.-P. Vandeborre, O. Colot, A new 3D-matching method of nonrigid and partially similar models using curve analysis, IEEE Trans. Pattern Anal. Mach. Intell. 33 (4) (2011) 852–858.
[61] D.V. Vranic, Desire: a composite 3D-shape descriptor, in: Proceedings of the IEEE International Conference on Multimedia and Expo, 2005, pp. 962–965.
[62] M. Liu, B.C. Vemuri, S. ichi Amari, F. Nielsen, Shape retrieval using hierarchical total bregman soft clustering, IEEE Trans. Pattern Anal. Mach. Intell. 34 (12) (2012) 2407–2419.
[63] H. Tabia, H. Laga, D. Picard, P.-H. Gosselin, Covariance descriptors for 3D shape matching and retrieval, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 4185–4192.
[64] H. Tabia, H. Laga, Covariance-based descriptors for efficient 3D shape matching, retrieval, and classification, IEEE Trans. Multimed. 17 (9) (2015) 1591–1603.
[65] P. Papadakis, I. Pratikakis, T. Theoharis, G. Passalis, S.J. Perantonis, 3D object retrieval using an efficient and compact hybrid shape descriptor, in: Proceedings of the Eurographics Workshop on 3D object retrieval, 2008, pp. 9–16.
[66] P. Papadakis, I. Pratikakis, T. Theoharis, S.J. Perantonis, Panorama: a 3D shape descriptor based on panoramic views for unsupervised 3D object retrieval, Int. J. Comput. Vis. 89 (2–3) (2010) 177–192.
[67] T. Furuya, R. Ohbuchi, Fusing multiple features for shape-based 3D model retrieval, in: Proceedings of the British Machine Vision Conference, 2014.
[68] Y. Fang, J. Xie, G. Dai, M. Wang, F. Zhu, T. Xu, E. Wong, 3D deep shape descriptor, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 2319–2328.
[69] J. Xie, Y. Fang, F. Zhu, E. Wong, Deepshape: deep learned shape descriptor for 3D shape matching and retrieval, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1275–1283.
[70] L. Bai, L. Cui, Y. Wang, X. Jin, X. Bai, E.R. Hancock, Shape classification with a vertex clustering graph kernel, in: Proceedings of the International Conference on Pattern Recognition, 2016, pp. 2634–2639.
[71] X. Tian, D. Tao, Y. Rui, Sparse transfer learning for interactive video search reranking, ACM Trans. Multimed. Comput. Commun. Appl. 8 (3) (2012) 26.

Song Bai received the B.S. degree in Electronics and Information Engineering from Huazhong University of Science and Technology (HUST), Wuhan, China in 2013. He is currently working toward the Ph.D. degree in the School of Electronic Information and Communications, HUST. His research interests include shape geometry, image classification, object retrieval, person re-identification and deep learning.

Shaoyan Sun received the Ph.D. degree in electronic engineering and information science from University of Science and Technology of China (USTC), China, in 2017. His research interests include multimedia information retrieval and computer vision.

Xiang Bai received the B.S., M.S., and Ph.D. degrees from the Huazhong University of Science and Technology (HUST), Wuhan, China, in 2003, 2005, and 2009, respectively, all in electronics and information engineering. He is currently a Professor with the School of Electronic Information and Communications, HUST. He is also the Vice-director of the National Center of Anti-Counterfeiting Technology, HUST. His research interests include object recognition, shape analysis, scene text recognition and intelligent systems.

Zhaoxiang Zhang received the B.S. degree in electronic science and technology from the University of Science and Technology of China (HUST), Hefei, China, in 2004, and the Ph.D. degree from the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2009. In 2009, he joined the School of Computer Science and Engineering, Beihang University, Beijing, China, as an Assistant Professor from 2009 to 2011, an Associate Professor from 2012 to 2015, and as the Vice-Director of the Department of Computer Application Technology from 2014 to 2015. In 2015, he returned to the Institute of Automation, Chinese Academy of Sciences, as a Full Professor. His current research interests include computer vision, pattern recognition, machine learning, and brain-inspired neural network and brain-inspired learning. Prof. Zhang is the Associate Editor or Guest Editor of some internal journals, like Neurocomputing, Pattern Recognition Letters, and IEEE ACCESS.

Qi Tian is currently a Full Professor in the Department of Computer Science, the University of Texas at San Antonio (UTSA). He was a tenured Associate Professor from 2008 to 2012 and a tenure-track Assistant Professor from 2002 to 2008. During 2008–2009, he took one-year Faculty Leave at Microsoft Research Asia (MSRA) as Lead Researcher in the Media Computing Group. Dr. Tian received his Ph.D. in ECE from University of Illinois at Urbana-Champaign (UIUC) in 2002 and received his B.E. in Electronic Engineering from Tsinghua University in 1992 and M.S. in ECE from Drexel University in 1996, respectively. Dr. Tian's research interests include multimedia information retrieval, computer vision, pattern recognition and bioinformatics and published over 400 refereed journal and conference papers. He was the co-author of a Best Paper in ACM ICMR 2015, a Best Paper in PCM 2013, a Best Paper in MMM 2013, a Best Paper in ACM ICIMCS 2012, a Top 10% Paper Award in MMSP 2011, a Best Student Paper in ICASSP 2006, and co-author of a Best Student Paper Candidate in ICME 2015, and a Best Paper Candidate in PCM 2007. Dr. Tian research projects are funded by ARO, NSF, DHS, Google, FXPAL, NEC, SALSI CIAS, Akiira Media Systems, HP, Blippar and UTSA. He received 2014 Research Achievement Awards from College of Science, UTSA. He received 2010 ACM Service Award. He is the associate editor of IEEE Transactions on Multimedia (TMM), IEEE Transactions on Circuits and Systems for Video Technology (TCSVT), Multimedia System Journal (MMSJ), and in the Editorial Board of Journal of Multimedia (JMM) and Journal of Machine Vision and Applications (MVA). Dr. Tian is the Guest Editor of IEEE Transactions on Multimedia, Journal of Computer Vision and Image Understanding, etc. Dr. Tian is a Fellow of IEEE.