Smooth Neighborhood Structure Mining on Multiple Affinity Graphs with Applications to Context-Sensitive Similarity

Song Bai[1(✉)], Shaoyan Sun[2], Xiang Bai[1], Zhaoxiang Zhang[3], and Qi Tian[4]

[1] Huazhong University of Science and Technology, Wuhan, China
{songbai,xbai}@hust.edu.cn
[2] University of Science and Technology of China, Hefei, China
sunshy@mail.ustc.edu.cn
[3] CAS Center for Excellence in Brain Science and Intelligence Technology, CASIA, Beijing, China
zhaoxiang.zhang@ia.ac.cn
[4] University of Texas at San Antonio, San Antonio, USA
qi.tian@utsa.edu

Abstract. Due to the ability of capturing geometry structures of the data manifold, diffusion process has demonstrated impressive performances in retrieval task by spreading the similarities on the affinity graph. In view of robustness to noise edges, diffusion process is usually *localized*, *i.e.*, only propagating similarities via neighbors. However, selecting neighbors smoothly on graph-based manifolds is more or less ignored by previous works. 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. Exhausted experimental results and comparisons against other algorithms manifest the effectiveness of the proposed algorithm.

Keywords: Diffusion process · Image/shape retrieval · Affinity graph

1 Introduction

In recent years, context-sensitive similarity has attracted much attention due to its superior performances in image/shape retrieval. These approaches have a very diverse nomenclature, such as contextual dissimilarity measure [1], graph transduction [2–4], affinity learning [5,6], ranking list comparison [7–9], re-ranking [10–12]. However, their inherent principle is almost the same, that is, the similarity between two images can be measured more accurately by taking the

B. Leibe et al. (Eds.): ECCV 2016, Part II, LNCS 9906, pp. 592–608, 2016.
DOI: 10.1007/978-3-319-46475-6_37

underlying manifold structure into account. In order to specify the differences between them systematically, Donoser *et al.* [13] provide a generic framework called Diffusion Process and a thorough comparison of most aforementioned algorithms experimentally. Diffusion process is operated on affinity graph, with nodes representing images and edge weights denoting pairwise similarities. This affinity graph actually defines a data manifold implicitly, and the similarities are diffused along the geodesic path of the manifold.

As one of the most important conclusions quoted from [13], 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 [14], 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 with short geodesic distances along the manifold. As a result, the nodes that diffusion process selects to spread the similarities on the affinity graph are usually the neighbors of the query that have small dissimilarities with it. So, it is of great importance to construct robust neighborhood structures so that diffusion process is performed in a proper way.

The simplest way to establish the neighborhood structure is k-nearest neighbors (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 [15] (also named as reciprocal kNN in [16,17]). As extensively proven in [15], kNN is prone to including noise edges and nodes, thus leading to unsatisfactory retrieval performances. To overcome its defect, Dominant Neighbors (DN) is proposed in [5] based on the analysis of dominant sets, and Consensus of kNN (CN) is proposed in [18] 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 [19], and later emphasized again in [13]. 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 more affinity graphs are given, the difficulties of constructing neighborhood structures lie in two aspects, *i.e.*, the way to determine the weights of different affinity graphs and the way to aggregate the neighborhood structures produced by them. Both issues are quite difficult if no prior knowledge is available. Of course, one can use a linear combination of multiple affinity graphs with equal weights. However, as demonstrated in Sect. 4, it is a suboptimal solution since the complementary nature among them is neglected.

In this paper, we propose an algorithm called Smooth Neighborhood (SN) specifically for neighborhood structure mining. Apart from previous works, our primary contributions can be divided into three parts: (1) SN enables the neighbor selection to vary smoothly along the data manifold, thus the local similarity can be sufficiently reflected in the selection of neighbors. (2) SN is suitable to deal with more than one affinity graph. It learns the shared neighborhood structure and the importance of multiple affinity graphs in a unified framework. Therefore the neighborhood aggregation and weight learning 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.

2 Related Work

Tremendous developments on context-sensitive similarities advance image/shape retrieval remarkably. A family of algorithms called diffusion process is proposed in the literature, such as Graph Transduction [2], Locally Constrained Diffusion Process (LCDP) [19], Locally Constrained Mixed Diffusion (LCMD) [20], Tensor Product Graph Diffusion (TPG) [5], Shortest Path Propagation (SSP) [3], Graph-PageRank [11], *etc.* In the survey paper [13], most of these approaches are elegantly summarized in a unified framework.

As shown in [13], a proper selection of neighbors ensures the diffusion process to work well in real cases. However, most variants of diffusion process use k-nearest neighbors (kNN) rule for its simplicity. Although [13] also uses kNN rule, it points out that it is still an open issue to select a reasonable local neighborhood. Related with this task, there are two representative algorithms recently, *i.e.*, Dominant Neighbors [5] and Consensus of kNN [18].

DN borrows the idea of dominant set proposed in [21] 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. Then an indicator vector is defined for each image to measure the probability of other images being its true neighbors. The indicator is subsequently learned by replicator equation [22].

Although DN achieves some improvements on retrieval performances, it still has some severe disadvantages. For example, it is prone to getting stuck at wrong local optima. In [18], 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. It is demonstrated sufficiently that CN can achieve quite stable performances than DN, especially with larger neighborhood size.

As a smooth operator to preserve the local structure of data manifold, graph Laplacian has been applied to various computer vision applications, such as feature coding [23], image annotation [24], semi-supervised learning [25]. Our approach, called Smooth Neighborhood (SN), is essentially based on the use of graph Laplacian. It interprets the procedure of neighbor selection in a probabilistic manner similar to DN.

3 Proposed Method

Given a collection of images $X = \{x_1, x_2, \ldots, x_N\}$, we can construct an undirect graph $\mathcal{G} = (X, W)$, where the vertices of the graph are images and $w_{ij} \in W$ measures the strength of the edge linking x_i and x_j. The problem needed to solve now is discovering a neighborhood set with high confidences for a given vertex $x_i \in X$.

As analyzed above, neither the simplest k-nearest neighbors (kNN) rule nor some more advanced algorithms (*e.g.*, [5,18,26]) cannot satisfy the manifold assumption. To remedy this, we propose a robust algorithm to select neighbors in an unsupervised way, formally defined as

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

where $Y_i = [y_{i1}, y_{i2}, \ldots, y_{iN}] \in \mathbb{R}^{1\times N}$ 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 [5]. $I_i \in \mathbb{R}^{1\times N}$ is the i-th row of an identity matrix I, indicating that x_i initializes itself as its nearest neighbor.

As can be seen from Eq. (1), we hold two assumptions. The left term emphasizes that the selection of neighbors should be smooth along the underlying manifold structure, *i.e.*, nearby points (large w_{ij}) should yield similar neighbors (small distance between Y_i and Y_j). 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 $\mu > 0$, and it should be 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. (1) to describe the importance of graphs, thus leading to our final objective function

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

where $\alpha = \{\alpha^{(1)}, \alpha^{(2)}, \ldots, \alpha^{(M)}\}$ is the weight of affinity graphs, and $\gamma > 1$ controls the weight distribution across multiple affinity graphs.

Three noteworthy comments should be made here. First, weight learning procedure in Eq. (2) 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 regard to α. Thus one can easily tune the weight distribution of these affinity graphs by varying γ.

Second, one may note that we do not set M indicator functions $Y_i^{(v)}$ corresponding with different affinity graphs $\mathcal{G}^{(v)}$. Instead, only a shared indicator function Y_i is utilized for node x_i. Such a setup has an inborn advantage that the consensus information among these affinity graphs can be exploited. Also the subsequent fusion of M indicator functions is avoided. Only one indicator function can be directly attained, though more affinity graph are used. Last, Eq. (2) is equivalent to Eq. (1) when $M = 1$. Therefore, Eq. (2) can be used directly on arbitrary number of input affinity graphs without modifications.

3.1 Optimization

For the sake of notation convenience, the objective function in Eq. (2) can be re-written in matrix form as

$$\mathcal{J} = \sum_{v=1}^{M} {\alpha^{(v)}}^{\gamma} Tr(Y^{\mathrm{T}} L^{(v)} Y) + \mu \|Y - I\|_F^2, \tag{3}$$

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

As we can see from Eq. (3), there are two variables to determine, *i.e.*, the indicator Y and the weight α. Hence, we decompose it into two sub-problems, then adopt an alternative way to solve the optimization problem iteratively.

Fix $\boldsymbol{\alpha}$, Update $\boldsymbol{Y}$. To get the optimal solution of this sub-problem, we compute the partial derivative of $\mathcal{J}$ with regard to Y as

$$\frac{\partial \mathcal{J}}{\partial F} = 2 \sum_{v=1}^{M} {\alpha^{(v)}}^{\gamma} L^{(v)} Y + 2\mu(Y - I). \tag{4}$$

Setting Eq. (4) to zero, the closed-form solution of F can be derived as

$$Y = \mu \left(\sum_{v=1}^{M} {\alpha^{(v)}}^{\gamma} L^{(v)} + \mu I \right)^{-1}. \tag{5}$$

Since graph Laplacian matrix is known to be positive semi-definite, we can easily derive that $\sum_{v=1}^{M} {\alpha^{(v)}}^{\gamma} L^{(v)}$ is also positive semi-definite. As a result, $\sum_{v=1}^{M} {\alpha^{(v)}}^{\gamma} L^{(v)} + \mu I$ is invertible as long as $\mu > 0$.

Fix $\boldsymbol{Y}$, Update $\boldsymbol{\alpha}$. In order to minimize Eq. (2) with regard to the graph weight $\alpha^{(v)}$, we utilize Lagrange Multiplier Method. Taking the constraint $\sum_{v=1}^{M} \alpha^{(v)} = 1$ into consideration, the Lagrange function of $\mathcal{J}$ is

$$L(\mathcal{J},\lambda)=\sum_{v=1}^{M}{\alpha^{(v)}}^{\gamma}Tr(Y^{\mathrm{T}}L^{(v)}Y)+\mu\|Y-I\|_F^2-\lambda\left(\sum_{v=1}^{M}\alpha^{(v)}-1\right), \tag{6}$$

whose partial derivatives with respect to $\alpha^{(v)}$ and λ are

$$\begin{cases}\dfrac{\partial L(\mathcal{J},\lambda)}{\partial\alpha^{(v)}}=\gamma{\alpha^{(v)}}^{(\gamma-1)}Tr(Y^{\mathrm{T}}L^{(v)}Y)-\lambda,\\ \dfrac{\partial L(\mathcal{J},\lambda)}{\partial\lambda}=-\sum\limits_{v=1}^{M}\alpha^{(v)}+1.\end{cases} \tag{7}$$

Note that in this sub-problem, $\|Y-I\|_F^2$ is a constant, and we can omit it directly.

By setting the two derivatives in Eq. (7) to zero simultaneously, the Lagrange multiplier λ is eliminated and the optimal solution of $\alpha^{(v)}$ is obtained finally as

$$\alpha^{(v)}=\frac{\left(Tr(Y^{\mathrm{T}}L^{(v)}Y)\right)^{\frac{1}{1-\gamma}}}{\sum_{v'=1}^{M}\left(Tr(Y^{\mathrm{T}}L^{(v')}Y)\right)^{\frac{1}{1-\gamma}}}. \tag{8}$$

For clarification, we summarize the whole procedure of optimization in Algorithm 1. 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 1. The pseudocode of smooth neighborhood.

Input:
$W^{(v)}\in\mathbb{R}^{N\times N}, 1\le v\le M$: the affinity matrices;
Two hyperparameters: γ and μ.
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. (5);
    Update the weight α using Eq (8);
  until *convergence*
  return Y

3.2 Remarks

Convergence. The convergence of the above optimization is guaranteed. For each sub-problem, we find the corresponding optimal solution. Consequently,

by solving two sub-problems alteratively, the objective value of Eq. (2) keeps decreasing monotonically. Meanwhile, since the objective function is lower bounded, the convergence of the proposed algorithm can be verified.

Affinity Initialization. In this paper, we need to specify the similarity matrix W. The most common way is to use Gaussian Kernel as

$$w_{ij} = \exp\left(-\frac{\|x_i - x_j\|_2^2}{\sigma_{ij}^2}\right), \tag{9}$$

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 affinity learning algorithms [2,19,20] use an adaptive kernel. For example, it is defined in [15] as $\sigma_{ij} = \sigma_i \sigma_j$, where $\sigma_i = \|x_i - x_{K(i)}\|_2$ and $K(i)$ is the index of the K-th 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 [13] on affinity learning. It is more helpful to figure out which part really works using a constant for affinity initialization.

Hyperparameters. There are two hyperparameters in our algorithm.

γ controls the weight distribution of multiple 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. Rich 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 parameters of 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 means that only x_i itself is selected as its neighbor finally.

Properties of Y. Y is a row-stochastic matrix (see Appendix for proof). This nice property satisfies the usual requirement for probabilistic algorithms (*e.g.*, dominant neighbors [5]) that for each node x_i, the sum of the probabilities of its neighbors is equal to 1.

4 Experiments

In this section, we will testify the validity of Smooth Neighborhood (SN) against other related algorithms, including Dominant Neighbors (DN) [5] and Consensus of kNN (CN) [18] on several visual retrieval tasks.

4.1 MPEG-7 Dataset

Following [5,13,18], the effectiveness of smooth neighborhood is first evaluated on MPEG-7 dataset [27]. It consists of 1,400 silhouette images divided into 70 categories, where each category has 20 shapes. The retrieval performance is measured by the bull's eye score, *i.e.*, the average recall of the top-40 returned candidates.

On this dataset, we implement four different shape similarity measures that are extensively used to learn the shape manifold in related literatures. They are Inner Distance Shape Context (IDSC) [28], Shape Context (SC) [29], Aspect Shape Context (ASC) [30] and Articulation-invariant Representation (AIR) [31]. The baseline performances of the four pairwise similarities are 85.40 %, 86.79 %, 88.39 % and 93.54 % respectively.

Qualitative Evaluation. Inspired by contextual re-ranking algorithms that leverage neighborhood set comparison directly for re-ranking (see [9,11,17]), we first adopt a simple and basic evaluation pipeline. 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 using Jaccard similarity as

$$S(x_q, x_p) = \frac{|\mathcal{N}(x_q) \cap \mathcal{N}(x_p)|}{|\mathcal{N}(x_q) \cup \mathcal{N}(x_p)|}, \tag{10}$$

where $|.|$ measures the cardinality of the input set. The motivation of Eq. (10) is straightforward, *i.e.*, if two images are similar, they tend to have extensive common neighbors.

In Fig. 1, we plot the retrieval performances of Eq. (10) 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. 1 shows that SN yields much smoother neighborhood structures than the other compared algorithms. Especially at larger K, the advantage of SN is more dramatic. It demonstrates clearly the benefit of exploring the local consistency in the proposed method. Since there are 20 shapes per category on MPEG-7 dataset, outliers are likely to be included when the neighborhood size is larger than 20. Nevertheless, the objective function in Eq. (2)

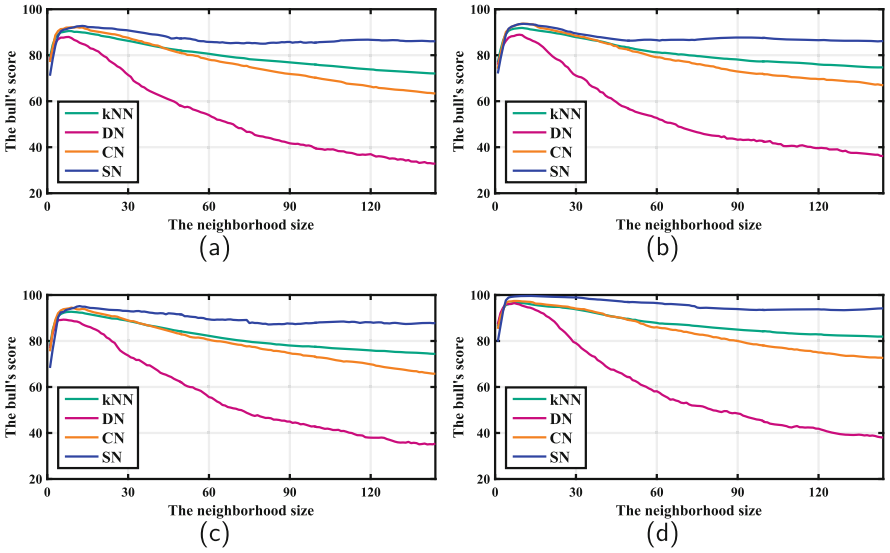

Fig. 1. The comparison using Eq. (10) on MPEG-7 dataset. The baseline similarities used are IDSC (a), SC (b), ASC (c) and IDSC+SC+ASC (d) respectively

regularizes that even if a relatively larger K is specified, the behaviour of selecting neighbors of nearby points are 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 reranking algorithms, where contextual information is described by neighborhood structures.

In [18], CN is also demonstrated to generate stable performances at larger neighborhood size. However, it is inspired by experimental observations, but lacks of 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 [18]. Second, we do not give a good enough initialization for similarities using adaptive Gaussian kernels as previous works [5] do.

When integrating multiple affinity graphs, the proposed 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. Moreover, 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.

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 [13] for a comprehensive summary). However, the convergence property of most these algorithms is not guaranteed, so the iteration has to be stopped at a "right" moment. Moreover, according to our experiments, some methods heavily rely on a proper initialization for pairwise similarities using adaptive Gaussian kernel as introduced in Sect. 3.2, *i.e.*, σ_{ij} should be different for each pair of shapes. If σ_{ij} is set to a constant as in this paper, these iterative algorithms, including Locally Constrained Diffusion Process (LCDP) [19], 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 diffusion process. 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 neighborhood analysis techniques.

Instead of using LCDP, we turn to a recent algorithm called Sparse Contextual Activation (SCA) [9], which is simple to implement and insensitive to parameter tuning. SCA is particularly suitable to evaluate neighborhood analysis techniques since it directly focuses on the usage of neighborhood structures by comparing two neighborhood sets in fuzzy set theory. It has two parameters to be fixed manually, *i.e.*, the parameter k_1 and k_2 determining the first-order and the second-order neighborhood size respectively. In our case, the two neighborhood sets can be obtained using kNN, DN, CN and the proposed SN.

Table 1. The comparison of neighborhood analysis algorithms on MPEG-7 dataset

Methods	IDSC	SC	ASC	AIR	IDSC+SC	IDSC+SC+ASC
SCA+kNN	90.07 %	93.23 %	94.34 %	99.97 %	98.20 %	97.61 %
SCA+DN	90.07 %	89.85 %	92.49 %	98.84 %	96.96 %	98.25 %
SCA+CN	92.70 %	93.97 %	95.25 %	99.85 %	97.61 %	97.86 %
SCA+SN	**93.52 %**	**95.25 %**	**95.98 %**	**100.00 %**	**99.25 %**	**99.81 %**

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). Consistent with previous analysis, the proposed SN achieves the best performances. Especially 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 combine SCA with SN to provide the retrieval performances of our method, if not specified otherwise.

Comparison with State-of-the-Art. In Table 2, we give a thorough comparison with other state-of-the-art algorithms. The results in the table are carefully

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

Descriptors	Methods	Bull's eye score
IDSC	Contextual dissimilarity measure (CDM) [1]	88.30 %
IDSC	Generic diffusion process (GDP)* [13]	90.96 %
IDSC	Index-based re-ranking [8]	91.56 %
IDSC	Graph transduction (GT) [2]	91.61 %
IDSC	Locally constrained diffusion process [19]	92.36 %
IDSC	RL-sim re-ranking [7]	92.62 %
IDSC	Shortest path propagation (SSP) [3]	93.35 %
IDSC	Mutual kNN graph (mkNN) [15]	93.40 %
IDSC	Sparse contextual activation (SCA) [9]	93.44 %
IDSC	**Smooth neighborhood (Ours)**	**93.52 %**
SC	Generic diffusion process (GDP)* [13]	92.81 %
SC	Graph transduction (GT) [2]	92.91 %
SC	Sparse contextual activation (SCA) [9]	95.21 %
SC	**Smooth neighborhood (Ours)**	**95.25 %**
ASC	Generic diffusion process (GDP)* [13]	93.95 %
ASC	Index-based re-ranking [8]	94.09 %
ASC	RL-sim re-ranking [7]	95.75 %
ASC	Locally constrained DP (LCDP) [19]	95.96 %
ASC	Tensor product graph (TPG) [5]	**96.47 %**
ASC	**Smooth neighborhood (Ours)**	95.98 %
IDSC+SC	Co-transduction [4]	97.72 %
IDSC+SC	Locally constrained mixed diffusion (LCMD) [20]	98.84 %
IDSC+SC	Sparse contextual activation (SCA) [9]	99.01 %
IDSC+SC	**Smooth neighborhood (Ours)**	**99.25 %**
AIR	Tensor product graph (TPG) [5]	99.99 %
AIR	Generic diffusion process (GDP) [13]	100.00 %
AIR	**Smooth neighborhood (Ours)**	**100.00 %**

classified according to the type of input baseline similarity. Since generic diffusion process [13] 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 in http://vh.icg.tugraz.at/index.php?content=topics/diffusion.php, thus marked with $\star$ 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, 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 example, TPG [5] 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 DN. Note that generic diffusion process [13] also reports 100 % accuracy 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 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 similarity is used.

Previous affinity fusion algorithms usually consider integrating only two similarity measures, *e.g.*, SC and IDSC. This is because most of them [4] are based on co-training framework that is only suitable to deal with two similarity measures. Our method potentially provides an alternative way of feature fusion at neighbor selection level. What is more important is that it does not limit 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 instance, by combining IDS, SC and ASC, SN can yield bull's eye score **99.81 %**. To our best knowledge now, it is the best performance on MPEG-7 dataset while AIR is not used.

4.2 Ukbench Dataset

We then evaluate the proposed approach on Ukbench image dataset [32]. It is comprised of $2,550$ objects and each object has 4 different view points 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, the perfect N-S score on this dataset is 4.

On this dataset, we implement 4 kinds of distance measures. They are

1. Bag of Words (BoW): SIFT descriptors are extracted at interest points produced by Hessian-affine detectors, and later converted to RootSIFT [33]. A codebook with $20k$ entries is learned with K-means on independent data. We follow the pipeline of Hamming embedding [34] 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 5-th convolutional layer and the 7-th 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 [11], 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 Sect. 4.1. $\sigma = 0.5$ is used for affinity initialization.

Extensive algorithms have reported their performances on Ukbench dataset. However, in Table 3, we only collect two kinds of results for comparison, *i.e.*, post-precessing 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.

Table 3. The N-S scores of different methods on 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 [35]	3.56
BoW (3.22)	Tensor product graph [5]	3.61
BoW (3.26)	Co-transduction [4]	3.66
BoW (3.50)	RNN re-ranking [16]	3.67
BoW (3.54)	Graph fusion [11]	3.67
BoW (3.33)	Contextual dissimilarity measure [1]	3.68
BoW (3.56)	Sparse contextual activation [9]	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 [11]	3.28
HSV (3.40)	Sparse contextual activation [9]	3.56
HSV (3.40)	**Smooth neighborhood (Ours)**	**3.56**
BoW (3.20,3.17,2.81)	Locally constrained mixed diffusion [20]	3.70
BoW (3.54), HSV (3.17)	Graph fusion [11]	3.77
BoW (3.54), HSV (3.17)	Graph fusion [12]	3.83
BoW (3.58), CNN (3.40), *etc.*	Query adaptive fusion [36]	3.84
BoW (3.56), HSV (3.40)	Sparse contextual activation [9]	3.86
BoW (3.13), CNN (3.87)	ONE [37]	3.89
BoW (3.57), CNN (3.44), *etc.*	**Smooth neighborhood (Ours)**	**3.98**

Graph Fusion [11] is a representative algorithm that integrates multiple affinity graphs by averaging the strength of edges with equally weights. It reports 3.77 N-S score by fusing local SIFT feature and holistic HSV color histogram, and later reports 3.83 in [12] by iteratively constructing the graph. In some sense, our method 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 [36], N-S score 3.84 is achieved by fusing five kinds of features. Using BoW and HSV as the input similarities, SCA [9] reports N-S score 3.86. Fusing BoW and CNN feature, ONE [37] achieves N-S score 3.89, which is the best performance to our knowledge. Besides those methods, [38] using query expansion reports NS score 3.67. [39] and [40] exploit local convolutional features, and report NS score 3.76 and 3.65 respectively. In this paper, we achieve the near perfect N-S score **3.98** by combining all the 4 similarities, including BoW, two CNN features and HSV. It outperforms the previous state-of-the-art remarkably.

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 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 neighbor and consensus of kNN. In particular, despite the perfect bull's eye score on MPEG-7 dataset, SN also achieves near perfect N-S score **3.98** on Ukbench dataset.

Since the proposed method focuses on neighborhood analysis, it can be potentially plunged into other retrieval systems (*e.g.*, RNN Re-ranking [16], kNN Re-ranking [35]) and other computer vision tasks (*e.g.*, image categorization [41], object detection, 3D shape recognition [42]), 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.

Acknowledgments. The authors would to thank Pedronette DCG. for providing the codes on MPEG-7 dataset. This work was supported in part by NSFC 61573160, NSFC 61429201 and China Scholarship Council. This work was supported in part to Dr. Qi Tian by ARO grants W911NF-15-1-0290 and Faculty Research Gift Awards by NEC Laboratories of America and Blippar.

Appendix

Equation (5) can be re-written as

$$Y = \left(I + \frac{\sum_{v=1}^{M} \alpha^{(v)^\gamma} L^{(v)}}{\mu} \right)^{-1}. \tag{11}$$

According to the Searle Set of Identity [43], Eq. (11) can be transformed into

$$Y = I - \frac{I}{\mu} \left(I + \frac{\sum_{v=1}^{M} \alpha^{(v)^\gamma} L^{(v)}}{\mu} \right)^{-1} \left(\sum_{v=1}^{M} \alpha^{(v)^\gamma} L^{(v)} \right). \tag{12}$$

Let $\mathbf{1}$ be a column vector with all elements equal to 1. We know the minimum eigenvalue of graph Laplacian matrix is 0, *i.e.*, $L^{(v)}\mathbf{1} = 0$. By postmultiplying both sides of Eq. (12) by $\mathbf{1}$, we can obtain $Y\mathbf{1} = \mathbf{1}$. The proof is complete.

References

1. Jegou, H., Schmid, C., Harzallah, H., Verbeek, J.J.: Accurate image search using the contextual dissimilarity measure. TPAMI **32**(1), 2–11 (2010)
2. Bai, X., Yang, X., Latecki, L.J., Liu, W., Tu, Z.: Learning context-sensitive shape similarity by graph transduction. TPAMI **32**(5), 861–874 (2010)
3. Wang, J., Li, Y., Bai, X., Zhang, Y., Wang, C., Tang, N.: Learning context-sensitive similarity by shortest path propagation. Pattern Recogn. **44**(10), 2367–2374 (2011)
4. Bai, X., Wang, B., Wang, X., Liu, W., Tu, Z.: Co-transduction for shape retrieval. In: Maragos, P., Paragios, N., Daniilidis, K. (eds.) ECCV 2010, Part III. LNCS, vol. 6313, pp. 328–341. Springer, Heidelberg (2010)
5. Yang, X., Latecki, L.J.: Affinity learning on a tensor product graph with applications to shape and image retrieval. In: CVPR, pp. 2369–2376 (2011)
6. Wang, B., Tu, Z.: Affinity learning via self-diffusion for image segmentation and clustering. In: CVPR, pp. 2312–2319 (2012)
7. Pedronette, D., Torres, R.: Image re-ranking and rank aggregation based on similarity of ranked lists. Pattern Recogn. **46**(8), 2350–2360 (2013)
8. Pedronette, D., Almeida, J., Torres, R.: A scalable re-ranking method for content-based image retrieval. Inf. Sci. **265**, 91–104 (2014)
9. Bai, S., Bai, X.: Sparse contextual activation for efficient visual re-ranking. TIP **25**(3), 1056–1069 (2016)
10. Chen, Y., Li, X., Dick, A., Hill, R.: Ranking consistency for image matching and object retrieval. Pattern Recogn. **47**(3), 1349–1360 (2014)
11. Zhang, S., Yang, M., Cour, T., Yu, K., Metaxas, D.N.: Query specific fusion for image retrieval. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part II. LNCS, vol. 7573, pp. 660–673. Springer, Heidelberg (2012)
12. Zhang, S., Yang, M., Cour, T., Yu, K., Metaxas, D.N.: Query specific rank fusion for image retrieval. TPAMI **37**(4), 803–815 (2015)
13. Donoser, M., Bischof, H.: Diffusion processes for retrieval revisited. In: CVPR, pp. 1320–1327 (2013)

14. Roweis, S.T., Saul, L.K.: Nonlinear dimensionality reduction by locally linear embedding. Science **290**(5500), 2323–2326 (2000)
15. Kontschieder, P., Donoser, M., Bischof, H.: Beyond pairwise shape similarity analysis. In: Zha, H., Taniguchi, R., Maybank, S. (eds.) ACCV 2009, Part III. LNCS, vol. 5996, pp. 655–666. Springer, Heidelberg (2010)
16. Qin, D., Gammeter, S., Bossard, L., Quack, T., van Gool, L.: Hello neighbor: accurate object retrieval with k-reciprocal nearest neighbors. In: CVPR, pp. 777–784 (2011)
17. Pedronette, D., Penatti, O., Torres, R.: Unsupervised manifold learning using reciprocal kNN graphs in image re-ranking and rank aggregation tasks. Image Vis. Comput. **32**(2), 120–130 (2014)
18. Premachandran, V., Kakarala, R.: Consensus of k-NNs for robust neighborhood selection on graph-based manifolds. In: CVPR, pp. 1594–1601 (2013)
19. Yang, X., Koknar-Tezel, S., Latecki, L.J.: Locally constrained diffusion process on locally densified distance spaces with applications to shape retrieval. In: CVPR, pp. 357–364 (2009)
20. Luo, L., Shen, C., Zhang, C., van den Hengel, A.: Shape similarity analysis by self-tuning locally constrained mixed-diffusion. TMM **15**(5), 1174–1183 (2013)
21. Pavan, M., Pelillo, M.: Dominant sets and pairwise clustering. TPAMI **29**(1), 167–172 (2007)
22. Pelillo, M.: Matching free trees with replicator equations. NIPS **2**, 865–872 (2002)
23. Gao, S., Tsang, I.W.H., Chia, L.T., Zhao, P.: Local features are not lonely-laplacian sparse coding for image classification. In: CVPR, pp. 3555–3561 (2010)
24. Wang, J., Chang, S.F., Zhou, X., Wong, S.T.: Active microscopic cellular image annotation by superposable graph transduction with imbalanced labels. In: CVPR, pp. 1–8 (2008)
25. Zhu, X., Ghahramani, Z., Lafferty, J., et al.: Semi-supervised learning using Gaussian fields and harmonic functions. In: ICML, pp. 912–919 (2003)
26. Kuang, Z., Li, Z., Fan, J.: Discovering the latent similarities of the KNN graph by metric transformation. In: ICMR, pp. 471–474 (2015)
27. Latecki, L.J., Lakämper, R., Eckhardt, U.: Shape descriptors for non-rigid shapes with a single closed contour. In: CVPR, pp. 424–429 (2000)
28. Ling, H., Jacobs, D.W.: Shape classification using the inner-distance. TPAMI **29**(2), 286–299 (2007)
29. Belongie, S., Malik, J., Puzicha, J.: Shape matching and object recognition using shape contexts. TPAMI **24**(4), 509–522 (2002)
30. Ling, H., Yang, X., Latecki, L.J.: Balancing deformability and discriminability for shape matching. In: Maragos, P., Paragios, N., Daniilidis, K. (eds.) ECCV 2010, Part III. LNCS, vol. 6313, pp. 411–424. Springer, Heidelberg (2010)
31. Gopalan, R., Turaga, P., Chellappa, R.: Articulation-invariant representation of non-planar shapes. In: Maragos, P., Paragios, N., Daniilidis, K. (eds.) ECCV 2010, Part III. LNCS, vol. 6313, pp. 286–299. Springer, Heidelberg (2010)
32. Nistér, D., Stewénius, H.: Scalable recognition with a vocabulary tree. In: CVPR, pp. 2161–2168 (2006)
33. Arandjelović, R., Zisserman, A.: Three things everyone should know to improve object retrieval. In: CVPR, pp. 2911–2918 (2012)
34. Jegou, H., Douze, M., Schmid, C.: Hamming embedding and weak geometric consistency for large scale image search. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 304–317. Springer, Heidelberg (2008)

35. Shen, X., Lin, Z., Brandt, J., Avidan, S., Wu, Y.: Object retrieval and localization with spatially-constrained similarity measure and k-NN re-ranking. In: CVPR, pp. 3013–3020 (2012)
36. Zheng, L., Wang, S., Tian, L., He, F., Liu, Z., Tian, Q.: Query-adaptive late fusion for image search and person re-identification. In: CVPR, pp. 1741–1750 (2015)
37. Xie, L., Hong, R., Zhang, B., Tian, Q.: Image classification and retrieval are one. In: ICMR, pp. 3–10 (2015)
38. Tolias, G., Jégou, H.: Visual query expansion with or without geometry: refining local descriptors by feature aggregation. Pattern Recogn. **47**(10), 3466–3476 (2014)
39. Paulin, M., Douze, M., Harchaoui, Z., Mairal, J., Perronin, F., Schmid, C.: Local convolutional features with unsupervised training for image retrieval. In: ICCV, pp. 91–99 (2015)
40. Babenko, A., Lempitsky, V.: Aggregating deep convolutional features for image retrieval. In: ICCV, pp. 1269–1277 (2015)
41. Bai, S., Bai, X., Liu, W.: Multiple stage residual model for image classification and vector compression. TMM **18**(7), 1351–1362 (2016)
42. Bai, S., Bai, X., Zhou, Z., Zhang, Z., Latecki, L.J.: Gift: a real-time and scalable 3D shape search engine. In: CVPR (2016)
43. Searle, S.R.: Matrix Algebra Useful for Statistics. Wiley-Interscience, Hoboken (1982)