Automatic Ensemble Diffusion for 3D Shape and Image Retrieval

Song Bai, *Student Member, IEEE*, Zhichao Zhou, Jingdong Wang
Xiang Bai, *Senior Member, IEEE*, Longin Jan Latecki, Qi Tian, *Fellow, IEEE*

***Abstract*—As a postprocessing procedure, diffusion process has demonstrated its ability of substantially improving the performance of various visual retrieval systems. Whereas, great efforts are also devoted to similarity (or metric) fusion, seeing that only one individual type of similarity cannot fully reveal the intrinsic relationship between objects. This stimulates a great research interest of considering similarity fusion in the framework of diffusion process (*i.e.*, fusion with diffusion) for robust retrieval.**

In this paper, we firstly revisit representative methods about fusion with diffusion, and provide new insights which are ignored by previous researchers. Then, observing that existing algorithms are susceptible to noisy similarities, the proposed Regularized Ensemble Diffusion (RED) is bundled with an automatic weight learning paradigm, so that the negative impacts of noisy similarities are suppressed. Though formulated as a convex optimization problem, one advantage of RED is that it converts back into the iteration-based solver with the same computational complexity as the conventional diffusion process. At last, we integrate several recently-proposed similarities with the proposed framework. The experimental results suggest that we can achieve new state-of-the-art performances on various retrieval tasks, including 3D shape retrieval on the ModelNet dataset, and image retrieval on the Holidays and Ukbench datasets.

***Index Terms*—Diffusion process, Object retrieval, 3D shape**

I. Introduction

Object retrieval is a fundamental yet hot topic in computer vision, which has attracted much attention for decades. Given a query instance, its target is to find objects sharing similar visual appearances with the query in a large database. For a long time, it is crucial to design discriminative representations, so that the metric defined on the representations can be robust to common deformations, such as rotation, occlusion, illumination, *etc*.

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, to Dr. Latecki by the NSF grant IIS-1302164. (*Corresponding author: Xiang Bai*)

S. Bai, Z. Zhou and X. Bai are with the School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan, China. S. Bai is also with the Inception Institute of Artificial Intelligence, Abu Dhabi, United Arab Emirates. (e-mail: songbai.site@gmail.com, {xbai, zzc}@hust.edu.cn).

J. Wang is with Microsoft Research, Beijing, China (e-mail: jingdw@microsoft.com).

L.J. Latecki is with the Department of Computer and Information Sciences, Temple University, 1925 N.12th Street, Philadelphia, PA 19122. (e-mail: latecki@temple.edu)

Q. Tian is with the Department of Computer Science, University of Texas at San Antonio, San Antonio, TX 78249-1604 USA (e-mail: qitian@cs.utsa.edu).

Conventionally, Bag-of-Words (BoW) is usually employed thanks to the design of local descriptors, such as scale invariant feature transform (SIFT) [1], color histogram [2] for images, inner distance shape context (IDSC) [3] and log-polar transform [4] for shapes, and curve analysis [5], covariance descriptor [6], [7], PANORAMA [8] for 3D models. In recent years, the rapid development of deep learning algorithms and GPU computing platforms has shifted the research attention to deep-learned features [9], [10], [11], [12], [13], [14], [15], which yield a remarkable performance boost over conventional handcrafted features.

Nevertheless, the underlying manifold structure is neglected when directly computing the pairwise similarity in the metric space. To this end, a re-ranking component called diffusion process (see [16] for a survey) is usually plugged as a postprocessing step to refine the search results. Diffusion process models the relationship between objects on a graph-based manifold, wherein similarity values are diffused along the geodesic path in an iterative manner until a kind of equilibrium is reached. That is to say, the learned similarity is not only determined by the visual representations of data points themselves, but also affected by the contextual distribution around them.

Meanwhile, different similarities generally focus on different aspects of objects. Thus, it may occur that two objects quite distant in one similarity space are close to each other in another space. The reason behind can be ascribed to the fact that different visual cues generally capture different visual properties and characteristic of objects. In Fig. 1, we show a query example on the Ukbench dataset [17] with four different similarities. As can be seen clearly, the first database image, a true positive, is successfully indexed using the similarity "NetVLAD" [18] and HSV color histogram, but fails to be correctly indexed by SPoC [9] and ResNet [19]. Moreover, the second database image, a false positive, is incorrectly returned only using NetVLAD but rejected by the other three similarities. As a consequence, enormous efforts are also devoted to similarity fusion for the sake of leveraging the complementary nature among those distinct feature modalities. For instance, a regularization-based feature selection algorithm is proposed in [20] to leverage both the sparsity and clustering properties of multiple features. Coupled Multi-Index (c-MI) [21] integrates SIFT descriptor and color descriptor into a multi-dimensional inverted index.

To inherit the property of manifold-preserving from diffusion process, fusion is usually reconsidered within the framework of diffusion process, which leads to a new methodology

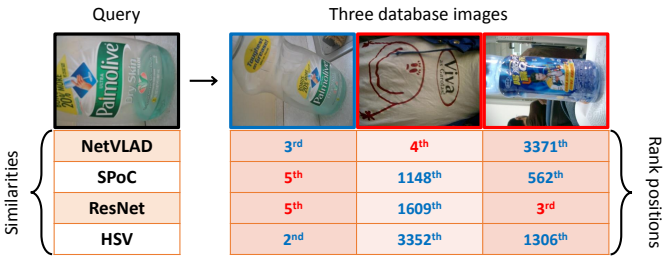


Fig. 1. The illustration of the complementarity among different similarities on the Ukbench dataset (details can be found in Sec. IV-B). The query, true positives and false positives are put in black, blue and red boxes, respectively. A prior knowledge on the dataset is that each category has 4 images. So, we mark those correct rank positions with blue color, and those false rank positions with red color.

called *Fusion with Diffusion*. Existing fusion with diffusion methods either utilize a naive solution by simply combining the edge weights of multiple affinity graphs (Sec. III-A and Sec. III-B), or consider a homogeneous fusion (Sec. III-C). However, most of them are susceptible to noisy similarities owing to the lack of an effective weight learning mechanism. To remedy this, we propose a novel fusion with diffusion method called Regularized Ensemble Diffusion (RED) in Sec. III-D. Compared with existing diffusion processes, the contributions of RED are two folds: i) RED is a theoretically sound and flexible approach to integrate multiple (more than two) similarities and learn their weights in the framework of diffusion process; ii) weights of similarities are learned in a totally unsupervised setting, whose essence is to seek for an optimal weight configuration to maximize the smoothness of multiple tensor-order graph manifolds.

The validity of those fusion with diffusion methods (both existing and newly proposed ones) is evaluated on various retrieval tasks. Benefiting from the progressive capacity of deep-based visual representations and handcrafted features, we are able to obtain new state-of-the-art performances with 3D shape retrieval on the ModelNet40 and ModelNet10 datasets [22], and with image retrieval on the Holidays [23] and Ukbench [17] datasets. Compared with [24], this paper further enhances the completeness of the summarized fusion with diffusion framework, by subdividing naive fusion into the early stage and the late stage, incorporated with both the sum rule and the product rule. It also provides more in-depth analysis of properties of the framework, particularly focusing on the advantages of the proposed RED. In addition, more comprehensive experimental comparisons and evaluations are given. It is observed that RED not only outperforms the baseline competitors, but also beats several representative and relevant algorithms [2], [25], [26] with the same input similarities.

The rest of paper is organized as follows. In Sec. II, we briefly review several representative variants of diffusion process. In Sec. III, we systematically summarize the formulations and the solutions of four different kinds of fusion with diffusion algorithms, including the proposed Regularized Ensemble Diffusion (RED). Experimental evaluations and comparisons are presented in Sec. IV and conclusions are given in Sec V.

II. Revisiting Diffusion Process

We begin with the preliminary facts about diffusion process on affinity graph in this section. Assume $\mathcal{G} = (X, W)$ is a weighted graph, where $X = \{x_1, x_2, \ldots, x_N\}$ is the set of vertices representing the data points, and the edge between x_i and x_j has weight $w_{ij} \in W$. Diffusion process learns a more faithful similarity $A \in \mathbb{R}^{N\times N}$ via iteration. According to the taxonomy given in [16], variants of diffusion process primarily differ in the definition of transition matrix and the updating scheme. Below we review some representative variants which can be deemed to operate on a tensor product graph, which is testified superior to other variants in [16].

Setting D as a diagonal matrix with elements $D_{ii} = \sum_{j'=1}^{N} W_{ij'}$, the transition matrix can be defined as $P = D^{-1}W$. Afterwards, Locally Constrained Diffusion Process (LCDP) [27] propagates the similarities via

$$A^{(t+1)} = PA^{(t)}P^{\mathrm{T}}, \tag{1}$$

where superscript t is the number of iterations. Since LCDP cannot guarantee the convergence, the iteration has to be stopped manually.

Apart from LCDP, Tensor Product Graph (TPG) diffusion [28] is proven to reach convergence after a sufficient number of iterations with the updating scheme

$$A^{(t+1)} = PA^{(t)}P^{\mathrm{T}} + I, \tag{2}$$

where $I \in \mathbb{R}^{N\times N}$ is the identity matrix.

Meanwhile, there are some other kinds of diffusion process, such as Manifold Ranking [29], Graph Transduction (GT) [30], Self Diffusion (SD) [31], Self-smoothing Operator (SSO) [32], Contextual Dissimilarity Measure (CDM) [33], *etc*. On the other hand, diffusion process also has close relationships with spatial verification [34], [35], and query expansion [36].

III. Fusion with Diffusion

Different from diffusion process that works with only one affinity graph, fusion with diffusion can tackle $M \geq 2$ affinity graphs $\mathcal{G}^v = (X, W^v)_{v=1}^{M}$ simultaneously. As above, our target is to learn the new similarity $A \in \mathbb{R}^{N\times N}$ which 1) captures the geometry of the underlying manifolds, and 2) leverages the complementarity among multiple visual features.

Since most fusion with diffusion methods stem from certain variants of diffusion process, we will use a new formulation of tensor product diffusion [37], [38], defined as

$$A^{(t+1)} = \alpha SA^{(t)}S^{\mathrm{T}} + (1-\alpha)I, \tag{3}$$

to keep consistency, where $S = D^{-1/2}WD^{-1/2}$ and $\alpha \in (0, 1)$ is a trade-off constant.

A. Naive Early Fusion

The most straightforward solution is to use a linear combination of multiple similarities. According to the type of fused similarities, naive fusion can be coarsely divided into two parts: naive early fusion and naive late fusion.

Naive early fusion simply averages the input similarities. Specifically, the transition matrix used in Eq. (3) is computed as

$$S = \frac{1}{M}\sum_{v=1}^{M} S^v, \tag{4}$$

where S^v is the transition matrix of the v-th affinity graph. Subsequently, a standard diffusion process is applied. This fusion strategy is extensively investigated in Locally Constrained Mixed Diffusion (LCMD) [39], Graph Fusion [2], [40], Yang *et al*. [41], *etc*.

Besides the sum rule employed in Eq. (4), product rule is also commonly used in data fusion. One of its merits against the sum rule is that the input data do not need to undergo a careful normalization to ensure the same scale of data distribution. In our specific scenario, the transition matrix can be computed as

$$S = \prod_{v=1}^{M} S^v, \tag{5}$$

where such a multiplication between matrices is known as the Hadamard product used in the matrix theory. From this perspective of view, naive early fusion with product rule can be regarded as a kind of similarity learning on the Hadamard product graph.

B. Naive Late Fusion

Naive late fusion chooses to average the learned similarities by diffusion process, instead of the input similarities. That is to say, for each affinity graph $\mathcal{G}^v$, the standard diffusion process in Eq. (3) is applied to obtain A^v. Then, the combined similarity A is computed as

$$A = \frac{1}{M}\sum_{v=1}^{M} A^v. \tag{6}$$

This fusion paradigm is used by Rank Diffusion [42], RL-Sim Re-ranking [43], Reciprocal kNN Graph Learning [44], *etc*. Naive late fusion also shares some common parts with ranking aggregation (*e.g.*, Contextual Dissimilarity Measure [33]), where multiple ranking lists generated by multiple cues are merged into one.

Accordingly, naive late fusion can be also implemented using the product rule, as

$$A = \prod_{v=1}^{M} A^v. \tag{7}$$

Though simple to implement, both naive early fusion and naive late fusion totally ignore the correlations among different similarities. Moreover, it is sensitive to noisy similarities, since it cannot adaptively decrease the weights of noise. Note that there exist some heuristic ways of weight learning here [41], [25].

C. Tensor Product Fusion

Another typical fusion strategy is tensor product fusion. Zhou *et al*. [45], [46] generalize TPG diffusion process [28] to deal with two similarities, where the high-order graph is built by computing the tensor product of two distinct affinity graphs.

By slightly adjusting the transition matrix in Eq. (3), we formulate the tensor product graph fusion as

$$A^{(t+1)} = \alpha S^{(2)} A^{(t)} {S^{(1)}}^{\mathrm{T}} + (1-\alpha) I, \tag{8}$$

where $S^{(1)}$ and $S^{(2)}$ are the transition matrices associated with two similarities, respectively.

It is easy to prove (see below) that after a sufficient number of iterations, Eq. (8) converges to

$$A = \lim_{t\to\infty} A^{(t)} = (1-\alpha)\mathrm{vec}^{-1}\left((I-\alpha\mathbb{S})^{-1}\tilde{I}\right), \tag{9}$$

where $\mathbb{S} \in \mathbb{R}^{N^2\times N^2}$ is the Kronecker product of $S^{(1)}$ and $S^{(2)}$, *i.e.*, $\mathbb{S} = S^{(1)} \otimes S^{(2)}$, and I is the identity matrix of the appropriate size. $\mathrm{vec}(\cdot)$ is the vectorization of the input matrix by stacking its columns one by one, and its inverse function is $\mathrm{vec}^{-1}(\cdot)$. To simplify the notation, we define $\tilde{Y} = \mathrm{vec}(Y)$ for any input matrix Y.

The proof can be as follows. By applying $\mathrm{vec}(\cdot)$ to both sides of Eq. (8), we obtain

$$\begin{aligned}\tilde{A}^{(t+1)} &= \alpha\mathbb{S}\tilde{A}^{(t)} + (1-\alpha)\tilde{I},\\ &= (\alpha\mathbb{S})^t\tilde{A}^{(1)} + (1-\alpha)\sum_{i=0}^{t-1}(\alpha\mathbb{S})^i\tilde{I}.\end{aligned} \tag{10}$$

It is known that the spectral radius of both $S^{(1)}$ and $S^{(2)}$ are no larger than 1. Hence, based on the spectrum property of the Kronecker product, all the eigenvalues of $\mathbb{S} = S^{(1)} \otimes S^{(2)}$ are also in $[-1, 1]$. Considering that $0 < \alpha < 1$, we have

$$\lim_{t\to\infty}(\alpha\mathbb{S})^t = 0, \quad \lim_{t\to\infty}\sum_{i=0}^{t-1}(\alpha\mathbb{S})^i = (I-\alpha\mathbb{S})^{-1}. \tag{11}$$

Therefore, one can easily induce that

$$\lim_{t\to\infty}\tilde{A}^{(t+1)} = (1-\alpha)(I-\alpha\mathbb{S})^{-1}\tilde{I}, \tag{12}$$

which proves Eq. (9) after applying vec^{-1} to its both sides.

Though the iterative formulation of tensor product fusion is intensively exploited, it lacks theoretical explanations of how this kind of similarity fusion better captures the manifold structure. In this paper, we demonstrate that the limit of $A^{(t)}$ presented in Eq. (9) can be derived by solving

$$\begin{aligned}\min_{A}\frac{1}{2}\sum_{i,j,k,l=1}^{N} W_{ij}^{(1)}W_{kl}^{(2)}\Big(\frac{A_{ki}}{\sqrt{D_{ii}^{(1)}D_{kk}^{(2)}}} - \frac{A_{lj}}{\sqrt{D_{jj}^{(1)}D_{ll}^{(2)}}}\Big)^2\\ + \mu\sum_{k,i=1}^{N}(A_{ki}-I_{ki})^2,\end{aligned} \tag{13}$$

where $\mu = \frac{1}{\alpha} - 1 \in (0, +\infty)$ is a constant regularizer.

Using several basic coordinate transformations and algebraic operations, Eq. (13) can be converted to

$$\min_{\tilde{A}} \tilde{A}^{\mathrm{T}}(I-\mathbb{S})\tilde{A} + \mu\|\tilde{A}-\tilde{I}\|^2. \tag{14}$$

To derive the optimal solution of the above problem, one can set its partial derivative with respect to $\tilde{A}$

$$2(I-\mathbb{S})\tilde{A}+2\mu(\tilde{A}-\tilde{I}) \tag{15}$$

to zero, substitute $\mu=\frac{1}{\alpha}-1$, and obtain

$$\tilde{A}=(1-\alpha)(I-\alpha\mathbb{S})^{-1}\tilde{I}, \tag{16}$$

which is equivalent to Eq. (9) after applying vec^{-1} to both sides.

Eq. (13) consists of two terms. The first term measures the smoothness of the tensor product graph, indicating that if x_i is similar to x_j in the first similarity space, *i.e.*, large $W^{(1)}_{ij}$, and x_k is similar to x_l in the second similarity space, *i.e.*, large $W^{(2)}_{kl}$, then the learned similarities A_{ki} and A_{lj} should have a small difference. The second term suggests that the self-similarity I_{kk} should be preserved to a certain extent. It should be emphasized here that tensor product fusion is not addressing cross-domain learning, where data points from two different domains have different representations. Instead, it aims at solving similarity learning within only one individual domain, when each data point in this domain has two complementary representations.

Here, we make a key observation that is essential in this paper. Though formulated in an iterative model, tensor product fusion can be theoretically explained using an optimization framework. Its essence is to seek for an optimal configuration A that minimizes the objective value of Eq. (13), and consequently, A maximizes the smoothness of the joint graph manifold. While the iterative formulation in Eq. (8) can only integrate two similarity matrices, it is possible to ensemble any number of such matrices in an optimization framework. More importantly, as we will show below, it is possible to learn weights for these similarities so that the smoothness of those manifolds is maximized, while keeping the algorithmic complexity unchanged.

Tensor product fusion considers the complementary structures of two different affinity graphs. However, it evades the weight learning issue, so that its performances are easily deteriorated once one of the two affinity graphs involves some noisy edges. Furthermore, in general, $S^{(1)}\otimes S^{(2)}\neq S^{(2)}\otimes S^{(1)}$. That is to say, when fusing two similarities, the order of computing the Kronecker product makes a difference.

D. Regularized Ensemble Diffusion

Insofar as we can conclude, learning the weights of multiple similarities has not been treated seriously by most existing fusion with diffusion methods. Since retrieval task usually does not have labeled training data, we expect weight learning can be done in an unsupervised manner, possibly with fewer additional parameters. Inspired by [26], [47] where a weight learning paradigm can be exerted on affinity graphs to assist neighborhood structure mining, the proposed Regularized Ensemble Diffusion (RED) makes viable the automatic weight learning for fusion with diffusion. However, the key novelty of RED lies in three facts. 1) Instead of using an exponential weight learner as [26], [47], RED adopts a more robust weight learning paradigm with regularization. 2) Although formulated as an optimization problem, RED can be efficiently solved in the spirit of the standard iteration-based diffusion process. 3) RED inherits the ability of capturing high-order relationships from tensor product diffusion [16], so that it can learn a more discriminative similarity.

Let $\beta=\{\beta_1,\beta_2,\ldots,\beta_M\}$ with β_v $(1\leq v\leq M)$ being the weight of the v-th affinity graph, and a larger weight indicates a greater importance for a given visual feature. We formulate the weight learning for β and the affinity learning for A in a unified framework as

$$\begin{aligned}&\min_{\beta,A}\sum_{v=1}^{M}\beta_v H^v+\mu\sum_{k,i=1}^{N}(A_{ki}-I_{ki})^2+\frac{1}{2}\lambda\|\beta\|_2^2,\\ &\textit{s.t.}\ \ 0\leq\beta_v\leq 1,\ \sum_{v=1}^{M}\beta_v=1,\end{aligned} \tag{17}$$

where

$$H^v=\frac{1}{2}\sum_{i,j,k,l=1}^{N}W^v_{ij}W^v_{kl}\Big(\frac{A_{ki}}{\sqrt{D^v_{ii}D^v_{kk}}}-\frac{A_{lj}}{\sqrt{D^v_{jj}D^v_{ll}}}\Big)^2, \tag{18}$$

and $\lambda>0$ is the weight **r**egularizer, controlling the distribution of the learned weights. H^v measures the smoothness of the v-th tensor-order data manifold, parameterized by W^v. As Eq. (17) shows, if the v-th graph is non-smooth (large H^v), it will be assigned small weight β_v such that the objective value will decrease. $\|\beta\|_2^2$ is a penalty parameter which avoids only favoring the smoothest similarity without exploiting the complementary among multiple similarities.

At first glance, one may doubt that RED is relevant to the iterative diffusion processes [16] expatiated above. However, as we present below, its optimization heavily relates to an iterative solver. The optimization of Eq. (17) can be decomposed into two subproblems:

1) Update A, fix β: In this situation, $\frac{1}{2}\lambda\|\beta\|_2^2$ is a constant which can be omitted directly. With similar operations used in Sec. III-C, Eq. (17) can be transformed into

$$\min_{\tilde{A}}\sum_{v=1}^{M}\beta_v\tilde{A}^{\mathrm{T}}(I-\mathbb{S}^v)\tilde{A}+\mu\|\tilde{A}-\tilde{I}\|^2, \tag{19}$$

where $\mathbb{S}^v=S^v\otimes S^v\in\mathbb{R}^{N^2\times N^2}$ is the Kronecker product of the v-th transition matrix with itself.

Closed-form Solution. The partial derivative of Eq. (19) with respect to $\tilde{A}$ is

$$2\sum_{v=1}^{M}\beta_v(I-\mathbb{S}^v)\tilde{A}+2\mu(\tilde{A}-\tilde{I}). \tag{20}$$

Since Eq. (19) is convex with respect to $\tilde{A}$, one can directly obtain its closed-form solution by setting Eq. (20) to zero. Consequently, we get

$$\tilde{A}=(1-\sum_{v=1}^{M}\alpha_v)(I-\sum_{v=1}^{M}\alpha_v\mathbb{S}^v)^{-1}\tilde{I}, \tag{21}$$

where

$$\alpha_v=\frac{\beta_v}{\mu+\sum_{v'=1}^{M}\beta_{v'}}. \tag{22}$$

Finally, the optimal solution of this subproblem can be obtained as $A = \text{vec}^{-1}(\tilde{A})$.

Iteration-based Solver. However, the closed-form solution in Eq. (21) is computationally prohibited even for small graphs. As we have to calculate the inverse of matrix of size $N^2 \times N^2$, the required time complexity is $O(N^6)$! This naturally motivates us to seek for an efficient iteration-based solver like the standard diffusion processes.

As we will prove below, the solution in Eq. (21) can be recovered by running

$$A^{(t+1)} = \sum_{v=1}^{M} \alpha_v S^v A^{(t)} S^{v\mathrm{T}} + (1 - \sum_{v=1}^{M} \alpha_v) I \tag{23}$$

for a sufficient number of iterations with an arbitrary initialization of $A^{(1)}$.

In more detail, Eq. (23) can be vectorized to

$$\begin{aligned} \tilde{A}^{(t+1)} &= \mathbb{S}\tilde{A}^{(t)} + (1 - \sum_{v=1}^{M} \alpha_v)\tilde{I} \\ &= \mathbb{S}^t \tilde{A}^{(1)} + (1 - \sum_{v=1}^{M} \alpha_v) \sum_{i=0}^{t-1} \mathbb{S}^i \tilde{I}. \end{aligned} \tag{24}$$

where $\mathbb{S} = \sum_{v=1}^{M} \alpha_v \mathbb{S}^v$.

Similar to Eq. (10), we only need to prove that all the eigenvalues of $\mathbb{S}$ are in $(-1, 1)$. Since all the eigenvalues of $\mathbb{S}^v$ $(1 \leq v \leq M)$ are in $[-1, 1]$, the spectral radius of $\mathbb{S}$ is bounded by $\sum_{v=1}^{M} \alpha_v$. Considering $\mu > 0$ and $\beta_v > 0$, we have

$$\sum_{v=1}^{M} \alpha_v = \frac{\sum_{v'=1}^{M} \beta_{v'}}{\mu + \sum_{v'=1}^{M} \beta_{v'}} < 1. \tag{25}$$

Therefore,

$$\lim_{t\to\infty} \mathbb{S}^t = 0, \quad \lim_{t\to\infty} \sum_{i=0}^{t-1} \mathbb{S}^i = (I - \mathbb{S})^{-1}. \tag{26}$$

Then,

$$\lim_{t\to\infty} \tilde{A}^{(t+1)} = (1 - \sum_{v=1}^{M} \alpha_v)(I - \sum_{v=1}^{M} \alpha_v \mathbb{S}^v)^{-1} \tilde{I}. \tag{27}$$

The proof is complete.

By using the iteration-based solver, the time complexity of updating A is reduced from $O(N^6)$ to $O(N^3)$.

2) Update β, fix A: When A is fixed, Eq. (17) can be simplified to the following problem

$$\min_{\beta} \sum_{v=1}^{M} \beta_v H^v + \frac{1}{2}\lambda \|\beta\|_2^2, \ \textit{s.t.}\ 0 \leq \beta_v \leq 1, \ \sum_{v=1}^{M} \beta_v = 1. \tag{28}$$

It can be efficiently solved using coordinate descent.

In each iteration of the coordinate descent, two elements β_i and β_j are selected to be updated, while the others are fixed. Taking into account the Lagrange function for the constraint $\sum_{v=1}^{M} \beta_v = 1$, we have the following updating scheme

$$\begin{cases} \beta_i^* = \frac{\lambda(\beta_i + \beta_j) + (H^j - H^i)}{2\lambda}, \\ \beta_j^* = \beta_i + \beta_j - \beta_i^*. \end{cases} \tag{29}$$

Algorithm 1: Regularized Ensemble Diffusion.

Input:
M similarity matrices $\{W^{(v)}\}_{v=1}^{M} \in \mathbb{R}^{N\times N}$, λ, μ
Output:
The learned similarity A;
begin
  Initialize the weight $\beta^{(v)} = \frac{1}{M}$;
  repeat
    Update A using Eq. (23);
    Update β using Eq. (29);
  until *convergence*
  return A

The obtained β_i^* (or β_j^*) may violate the constraint $\beta_v \geq 0$. Hence, we set $\beta_i^* = 0$ if $\lambda(\beta_i + \beta_j) + (H^j - H^i) < 0$, and vice versa for β_j^*.

The above optimization procedure is guaranteed to converge. In each subproblem, we obtain its optimal solution. By solving two subproblems alternatively, the objective value of Eq. (17) keeps decreasing monotonically. In addition, the objective function is lower bounded by zero. Thus, the convergence can be verified.

Compared with the naive fusion and tensor product fusion, RED is robust to noisy features by adaptively tuning the weights β. More importantly, zero weights are allowed such that irrelevant graphs can be totally filtered out, which is impossible for the approach in [26]. The pseudocode of the derived ensemble diffusion is presented in Alg. 1. Note that as no prior knowledge is available to judge the discriminative power of different similarities in unsupervised retrieval, a natural initialization for the weight β_v is $\frac{1}{M}$.

3) Further Explanation of the Loss Function: We consider here a pathological case of input similarity $W^{(\triangle)}$ that makes the loss function of RED uninformative. This happens when $W^{(\triangle)} = I$ is defined as

$$W_{ij}^{(\triangle)} = \begin{cases} 1, & \text{if } \ i = j \\ 0, & \text{if } \ i \neq j, \end{cases} \tag{30}$$

which only contains self-similarity of each object.

Indeed, in this case, $H^{(\triangle)}$ is equal to 0 according to Eq. (18). Afterwards, RED will only favor the similarity $W^{(\triangle)}$ by setting $\beta_{(\triangle)} = 1$. In other words, all the other similarities are discarded. However, we emphasize that the goal of this paper is to learn a more faithful similarity from multiple pairwise similarities. Hence, the basic requirement is that we need to know exactly a meaningful relationship between objects. Unfortunately, $W^{(\triangle)}$ is meaningless in this sense, since it cannot depict the relationship between two objects. In this case, neither the proposed methods nor existing affinity learning algorithms can learn a meaningful similarity for retrieval.

In summary, although the proposed method can handle any types of similarities in terms of its basic theory, we do not encourage the users to deliberately force $W^{(\triangle)}$ to be the input, since it is an ill-defined similarity.

TABLE I
THE SUMMARY OF FUSION WITH DIFFUSION METHODS. M DENOTES THE NUMBER OF SIMILARITIES THAT CAN BE HANDLED. #WEIGHT DENOTES WHETHER WEIGHT LEARNING EXISTS. #NOISE DENOTES THE ROBUSTNESS TO NOISE. #COMPLEXITY DENOTES THE TIME COMPLEXITY. #PARAMETER DENOTES THE NUMBER OF ADDITIONAL PARAMETERS OVER THE STANDARD DIFFUSION PROCESS. #PERFORMANCE DENOTES THE EXPERIMENTAL PERFORMANCES.

Methods	M	#Weight	#Noise	#Complexity	#Parameter	#Performance
Naive Early Fusion	≥ 2	$\times$	Good	$O(t_d N^3)$	0	Bad
Naive Late Fusion	≥ 2	$\times$	Good	$O(t_d M N^3)$	0	Good
Tensor Product Fusion	$=2$	$\times$	Bad	$O\left(t_d M(M-1)N^3\right)$	0	Good
RED (ours)	≥ 2	$\surd$	Excellent	$O\left(T(t_d M N^3 + k^4 + t_a M^2)\right)$	1	Excellent

E. Discussions

We summarize the inherent differences between the considered fusion with diffusion methods in Table I.

As the table suggests, the biggest defect of tensor product fusion is that it can only deal with $M = 2$ similarities, limiting its promotion where multiple similarities ($M \geq 2$) are accessible. Secondly, RED is the most robust to noisy similarities owing to the weight learning mechanism, followed by naive fusion and tensor product fusion. The robustness of naive fusion comes from a statistical assumption that if two objects are similar in most similarity spaces, they are true matching pairs. Concerning naive fusion, we observe that sum rule is more capable than product rule in the framework of diffusion process. In particular, for naive early fusion, product rule totally fails. The reason is that naive early fusion with product rule sets an edge between two vertices if and only if all their similarities think the two vertices are connected. That is to say, once there exists a zero similarity value between two vertices, they will be disconnected since the edge weight is made to zero using the Hadamard product. See the section below for the experimental support.

The time complexity of naive early fusion (both sum rule and product rule) is the lowest. It only requires one diffusion step in Eq. (3), leading to a complexity of $O(t_d N^3)$, where t_d is the number of iterations in diffusion process. Naive late fusion needs $O(t_d M N^3)$ to finish M independent diffusion steps. Tensor product fusion has to select 2 similarities each time to fulfill the diffusion step. Thus, its entire time complexity is $O\left(t_d M(M-1)N^3\right)$. The complexity of RED has two parts. The first part is updating A in $O(t_d M N^3)$ via Eq. (23), while the second part is updating β. It seems that we need $O(N^4)$ to compute Eq. (18). However, as suggested in [16], [27], diffusion process is usually localized by propagating similarities only on k-nearest neighbor graphs, leading to $O(k^4)$ in this step. Then, the second part of RED requires $O(t_a M^2)$ (t_a is the number of iterations in coordinate descent) via Eq. (29). Considering the outer iteration number T in the alternating optimization, the final cost of RED is $O\left(T(t_d M N^3 + k^4 + t_a M^2)\right)$. Since $t_d, t_a, k, M \ll N$, we can conclude that the time complexity of all the above fusion with diffusion methods is dominated by $O(N^3)$, which is equivalent to the standard diffusion process [16].

Moreover, RED introduces an additional parameter λ, which is used to automatically tune the weight distribution of different input similarities. Consequently, it can handle diverse data distributions with a consistent performance improvement over other fusion with diffusion methods.

IV. EXPERIMENTS

In this section, we give a thorough evaluation of the considered fusion with diffusion methods on the ModelNet dataset [22] with 3D shape retrieval, the Holidays [23] and Ukbench [17] datasets with image retrieval.

Besides naive early fusion (NEF), naive late fusion (NLF) and tensor product fusion (TPF) discussed above, we also compare three newly-proposed image retrieval algorithms concerning either feature fusion or diffusion process, including

- Graph Fusion [2]: As a representative algorithm, Graph Fusion considers a naive fusion of multiple similarities with equal weights. To get multiple similarities directly comparable, the edge weights are expressed using the Jaccard coefficient of two neighborhood sets. Then, PageRank, which can be taken as a kind of diffusion process, is performed on the fused graph for re-ranking.
- Query-adaptive late fusion (QALF) [25]: QALF is a simple yet effective method for feature fusion at the score level. It calculates the weights of similarities by studying the L shape of ranking lists. The final similarity is obtained by a weighted combination without diffusion process.
- Smooth Neighborhood (SN) [26], [47]: Apart from affinity learning discussed in this paper, SN focuses on mining robust neighborhood structures on multiple affinity graphs. It imposes an exponential weight learner so that the weights of similarities are always larger than 0. Hence, it suffers from the fact that the negative effects of the noisy similarities cannot be entirely eliminated.

The baseline similarities and the evaluation protocols vary with the data modalities, which will be specified in Sec. IV-A and Sec. IV-B, respectively.

A. 3D Shape Retrieval

The proposed framework is firstly evaluated on 3D shape retrieval on the ModelNet dataset, which is a large-scale 3D shape repository, currently consisting of $151,128$ 3D CAD models in 662 object categories. Following [22], two subsets are used for evaluation, *i.e.*, ModelNet40, containing $12,311$ shapes divided into 40 object categories, and ModelNet10, containing $4,899$ shapes divided into 10 object categories. As for the experimental setup, we use the same training-testing split as [48], [49], and employ Area Under precision-recall Curve (AUC) and mean Average Precision (mAP) as evaluation metrics. The parameters are given as $k = 16$ in kNN graph, $\mu = 0.3$ for fastening self-similarity, and $\lambda = 19$ for both datasets.

TABLE II
THE PERFORMANCES (%) OF BASELINES ON THE MODELNET40 AND MODELNET10 DATASETS, RESPECTIVELY.

Methods	ModelNet40		ModelNet10	
	AUC	mAP	AUC	mAP
Vol. CNN	80.39	79.53	91.24	89.97
GIFT	77.19	76.52	88.97	87.98
ResNet	80.12	79.41	89.02	88.17
PANO.	45.10	44.52	62.37	61.47

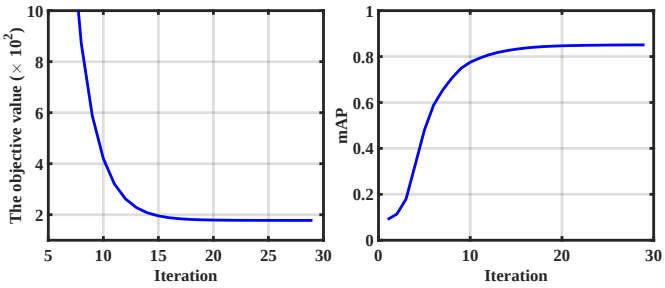


Fig. 2. The objective value and mAP (%) of tensor product fusion at each iteration on the ModelNet40 dataset. The two similarities fused are Volumetric CNN and GIFT.

TABLE III
THE MAP (%) COMPARISON OF TENSOR PRODUCT FUSION ON THE MODELNET40 DATASET. THE BEST AND THE WORST FUSING RESULTS ARE MARKED IN RED AND BLUE, RESPECTIVELY.

	Vol. CNN	GIFT	ResNet	PANO.
Vol. CNN	-	83.23	83.77	**67.16**
GIFT	83.15	-	85.08	69.15
ResNet	83.86	**85.12**	-	69.56
PANO.	69.29	71.08	72.32	-

TABLE IV
THE PERFORMANCE COMPARISON (%) OF DIFFERENT FUSION METHODS ON THE MODELNET40 DATASET. AS TENSOR PRODUCT FUSION CAN ONLY DEAL WITH TWO SIMILARITIES, ITS PERFORMANCES ARE GIVEN IN AN INTERVAL.

Methods	AUC	mAP
NEF-Product	72.45	71.38
NEF-Sum	85.26	84.55
NLF-Product	85.39	84.54
NLF-Sum	86.52	85.71
Tensor Product Fusion	68.56∼86.00	67.16∼85.12
QALF [25]	82.78	82.08
Graph Fusion [2]	83.82	83.09
SN [26]	84.15	83.20
RED (Ours)	**87.03**	**86.30**

Baselines. To obtain multiple similarities, we implemented 4 representative baselines. They are

- Volumetric CNN [50]: It is combined with multi-view Convolutional Neural Network (CNN), and uses 3DCNN with multi-oriented pooling to obtain shape representations.
- GIFT [49]: It is an elaborative search engine focusing on the scalability of 3D shape retrieval. We follow its pipeline by training an 8-layer CNN as the view feature extractor, and apply Hausdorff matching to activations of the $7th$ fully-connected layer;
- ResNet [19]: As a residual learning framework capable of training ultra deep networks, ResNet has shown outstanding performances on image classification and object detection. We introduce, for the first time, ResNet for 3D shape analysis. Here we finetune a 50-layer ResNet to extract view features, and utilize Hausdorff distance as [49] for shape matching;
- PANORAMA [8]: it is a classical shape descriptor, which is comprised of Discrete Fourier Transform and Discrete Wavelet Transform calculated on panoramic views.

The performances of those baselines are listed in Table II.

Analysis of Tensor Product Fusion. In Fig. 2, we plot the objective value of Eq. (13) and the retrieval performance at each iteration of diffusion process. As can be clearly seen, when similarities are propagated iteratively, the objective value keeps decreasing and the retrieval performance keeps increasing until reaching the equilibrium. Such an observation validates our new perspective about tensor product fusion in Sec. III-C, *i.e.*, the essence of the iterative solver of tensor product fusion is to recover a closed-form solution of an optimization problem, which measures the smoothness of the joint graph manifold.

Table III lists the retrieval performances of tensor product fusion by combining different pairs of the four similarities. Firstly, different orders of fusing the same two similarities yield different performances. Second, tensor product fusion totally fails when one of the two to-be-fused similarities is not discriminative enough. For instance, the fusion of Volumetric CNN and PANORAMA achieves mAP 67.16, significantly lower than the baseline mAP 79.53 achieved by Volumetric CNN.

Comparison of Fusion Methods. Table IV shows the comparison of different fusion methods. All the competitors are implemented using the same similarities as RED to ensure a fair comparison.

One can firstly observe that naive early fusion with product rule achieves inferior performances, even worse than the used baseline similarities. As analyzed in Sec. III-E, this is due to its extremely strict rule of constructing the affinity graph, which leads to the loss of informative edges. With the sum rule, the performance of naive early fusion is improved significantly, with the improvements of 12.80 in AUC and 13.17 in mAP. Nevertheless, naive early fusion still performs worse than naive late fusion, with either product rule or sum rule. Meanwhile, tensor product fusion (its highest AUC and mAP are 86.00 and 85.12, respectively) outperforms naive early fusion slightly and is comparable with naive late fusion, since it considers the complementary structures between two homogenous graphs.

Graph Fusion [2], QALF [25] and SN [26] are initially designed for image retrieval. However, considering that their ultimate goal is similarity learning or fusion, it is natural that they are not limited to image analysis, but also competent in 3D shape retrieval. In comparison with them, the superiority of RED firstly lies in the fact that we adopt a more robust weight learning paradigm with a theoretical guarantee than equal weights used by Graph Fusion, the exponential weight learner used by SN and the heuristic weight learning by QALF. More importantly, RED formulates the weight learning and the tensor-order affinity learning in a unified framework, which

TABLE V
THE LEARNED WEIGHTS β BY RED.

Datasets	Vol. CNN	GIFT	ResNet	PANO.
ModelNet10	0.281	0.332	0.387	0
ModelNet40	0.296	0.356	0.348	0

can efficiently output a more accurate search result.

As can be drawn from Table IV, RED achieves much better performances in both evaluation metrics. One desirable expectation is that the weight of PANORAMA should be smaller, since it leads to much lower retrieval performances. Table V presents the weights learned by RED. As can be seen, RED sets it to 0 on both ModelNet40 and ModelNet10 datasets. Since RED allows for zeros weights, ultra non-smooth graphs (PANORAMA in our case) can be totally eliminated. Therefore, RED can be adapted to more diverse situations even if considerable noise exists. These results also indicate that if more noisy similarities are fused, the performance difference between RED and other fusion methods will be more dramatic. We will further investigate this in Sec. IV-C, where simulated Gaussian noise is added.

Besides, it is observed that the weight of ResNet is larger than that of GIFT on the ModelNet40 dataset, but it is not the case on the ModelNet10 dataset. The reason is that the weights of similarities are learned in a data-driven way according to Eq. (17). That means if the data manifold is changed, the weight distribution will be changed accordingly. Thus, we can find that ResNet is more capable of handling the specific data distribution on the ModelNet40 dataset compared with GIFT, while GIFT is better on the ModelNet10 dataset.

More Weak Baselines. To further test the adaptation power of RED to noise, we introduce one more weak baseline similarity. For each 3D shape, SIFT descriptors [1] are extracted on depth images, which are encoded via VLAD [51] to form its vector representation. The mAP of the VLAD baseline is 45.91 on the ModelNet40 dataset, which is marginally higher than PANORAMA by only 1.39.

It is known that the similarities derived from local descriptors are generally invariant to the deformations, such as occlusion, rotation, *etc*. Therefore, although the VLAD baseline is not discriminative enough, it possesses the complementary nature to the holistic signatures used above. As a result, after fusing this weak baseline, the performance of RED is further improved from 86.30 (shown in Table IV) to mAP 86.50. Meanwhile, the weight of the VLAD baseline learned by RED is merely 0.023, which decreases the negative influence brought by its poor discriminative power. We envision that better performances can be achieved if more complementary similarities (*e.g.*, [52], [53], [11], [13], [54], [55]) are fused by the proposed algorithm.

Comparison with State-of-the-art. In Table VI, we give a comparison with all the representative methods which report retrieval performances on the ModelNet dataset[1].

The best-performing existing methods are Multi-view Convolutional Neutral Network (MVCNN) [48], GIFT [49] and Deep Local feature Aggregation Network (DLAN) [59]. Specifically, MVCNN proposes a view aggregation layer to produce a compact 3D shape descriptor. By using metric learning additionally, it reports mAP 78.90 on the ModelNet40 dataset. Meanwhile, GIFT introduces approximated Hausdorff distance for multi-view matching and Aggregated Contextual Activation for re-ranking. It reports mAP 81.94 on the ModelNet40 dataset. DLAN starts with extracting a set of local 3D signatures at multiple positions and scales from a 3D model. Those local signatures are then aggregated via a neural network to produce a rotation-invariant and compact feature per 3D model. Thanks to its ability in learning rotation-invariant representations, it leads to a considerable performance improvement on the ModelNet40 dataset, reporting mAP 85.00. In comparison, the proposed RED yields mAP 86.30 on the ModelNet40 dataset, outperforming MVCNN by 7.40, GIFT by 4.36 and DLAN by 1.30 percent, respectively. Besides, RED yields a new best performance on ModelNet10 dataset, that is, AUC 93.20 and mAP 92.15.

TABLE VI
THE PERFORMANCE COMPARISON (%) WITH THE STATE-OF-THE-ART ON THE MODELNET40 AND MODELNET10 DATASETS.

Methods	ModelNet40		ModelNet10	
	AUC	mAP	AUC	mAP
SPH [56]	34.47	33.26	45.97	44.05
LFD [57]	42.04	40.91	51.70	49.82
PANORAMA [8]	45.00	46.13	60.72	60.32
ShapeNets [22]	49.94	49.23	69.28	68.26
DeepPano [58]	77.63	76.81	85.45	84.18
MVCNN [48]	-	78.90	-	-
GIFT [49]	83.10	81.94	92.35	91.12
DLAN [59]	-	85.00	-	90.60
RED (ours)	**87.03**	**86.30**	**93.20**	**92.15**

Qualitative Evaluation. Fig. 3 presents two sample retrieval results for an additional evaluation on the ModelNet40 dataset. As can be clearly seen, RED outperforms the baseline similarities by a large margin. Particularly in Fig. 4b, one can observe that all the four baseline similarities fail with this query. However, by exploiting the complementary nature and the shared information among them, RED still improves the retrieval performance.

B. Image Retrieval

We also test the proposed methods on two well-known benchmark datasets for image retrieval, *i.e.*, Holidays dataset [23] and Ukbench dataset [17].

Holidays dataset contains $1,491$ personal photos, among which 500 images are used as queries. Most queries only have 1-2 ground-truth images, which makes the dataset very challenging for diffusion-based re-ranking methods. The retrieval performance is measured by mean Average Precision (mAP) over all the queries. Ukbench dataset consists of $10,200$ images, grouped into $2,550$ categories. Each image is taken as the query in turn and the rest images serve as the database. The evaluation metric is called N-S score, which counts the average recall of the top-4 ranked images. Thus, the perfect N-S score is 4. Since the scale and the category distribution of the two datasets are quite different, we set $k = 7$, $\lambda = 28$

[1]available at http://modelnet.cs.princeton.edu/.

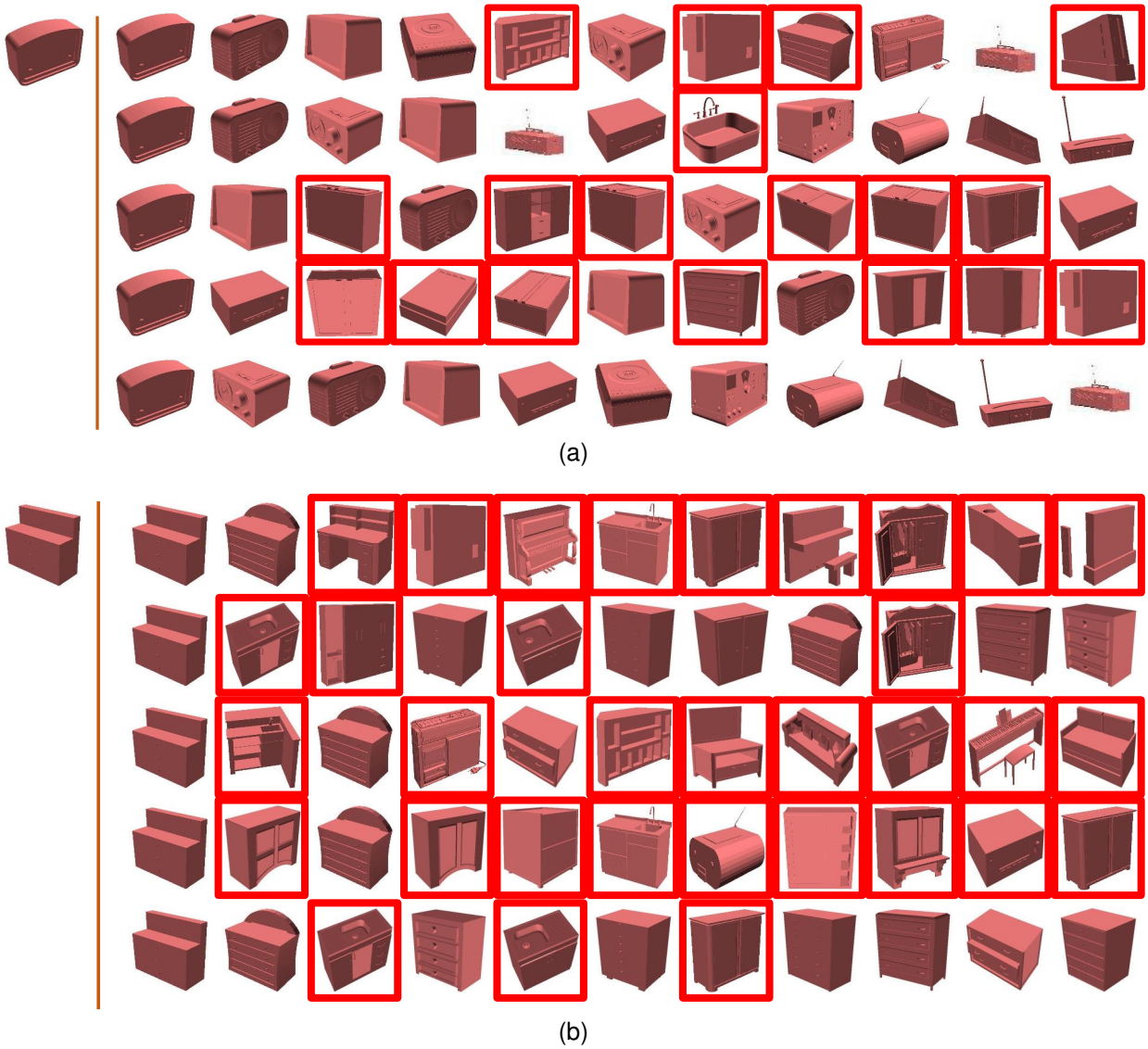


Fig. 3. The sample retrieval results on the ModelNet40 dataset, where the queries (shown in left) and their top-10 retrieval candidates are given. False positives are in red boxes. The retrieval results of the four baseline similarities, including Volumetric CNN, GIFT, ResNet and PANORAMA, are presented in the first four rows. The retrieval results of RED are presented in the fifth row.

on the Holidays dataset, and $k = 3$, $\lambda = 4K$ on the Ukbench dataset. μ is set to 0.08 on both datasets.

Baselines. Four baseline similarities are re-implemented:

- NetVLAD [18]: An end-to-end trained network which has a new generalized Vector of Locally Aggregated Descriptors (VLAD) [60] layer;
- SPoC [9]: A strategy based on the sum-pooling activations of convolutional layers of pretrained CNNs;
- ResNet [19]: The fully-connected layer of a pretrained 50-layer ResNet is used to extract holistic features;
- HSV color histogram: Following [2], [40], [21], we extract 1000-dimensional HSV color histograms ($20\times10\times5$ bins for H, S, V components).

Note that for deep features, the rotated version of Holidays dataset released in [61] is used.

Except for NetVLAD, all the extracted features are firstly square-rooted [51], and then L_2 normalized. Especially for SPoC, with such a square-root normalization, we obtain much higher performance than the original one reported in [9]. Table VII shows the performances of our implementation of the 4 baseline methods.

TABLE VII
THE PERFORMANCES OF BASELINES ON THE HOLIDAYS (MAP) AND UKBENCH (N-S SCORE) DATASETS, RESPECTIVELY. NOTE THAT SPOC ORIGINALLY REPORTS IN [9] MAP 80.2 ON THE HOLIDAYS DATASET AND N-S SCORE 3.65 ON THE UKBENCH DATASET, WHICH IS DIFFERENT FROM OUR IMPLEMENTATION OF SPOC DENOTED WITH SPOC*.

Datasets	NetVLAD	SPoC*	ResNet	HSV
Holidays	88.29	86.07	81.83	61.83
Ukbench	3.739	3.698	3.709	3.195

Comparison of Fusion Methods. Table VIII lists the detailed performances of tensor product fusion, and Table IX compares the results of different fusion methods.

In consistent with previous experiments, we observe that the proposed RED achieves the best performances due to

TABLE VIII
THE PERFORMANCES OF TENSOR PRODUCT FUSION ON THE HOLIDAYS (LEFT) AND UKBENCH (RIGHT) DATASETS. THE BEST AND THE WORST FUSING RESULTS ARE MARKED IN RED AND BLUE, RESPECTIVELY.

	NetVLAD	SPoC	ResNet	HSV
NetVLAD	-	92.36/3.871	91.85/3.874	88.24/**3.626**
SPoC	**92.46**/3.876	-	90.02/3.854	87.55/3.629
ResNet	91.85/**3.884**	90.09/3.861	-	85.44/3.629
HSV	87.77/3.680	87.33/3.685	**85.12**/3.682	-

TABLE IX
THE PERFORMANCE COMPARISON OF DIFFERENT FUSION METHODS ON THE HOLIDAYS AND UKBENCH DATASETS.

Methods	Holidays	Ukbench
NEF-Product	74.51	3.046
NEF-Sum	90.69	3.907
NLF-Product	90.23	3.668
NLF-Sum	90.92	3.936
Tensor Product Fusion	85.12∼92.46	3.626∼3.884
QALF [25]	88.31	3.846
Graph Fusion [2]	90.65	3.918
SN [26]	91.72	3.919
RED (ours)	**93.32**	**3.938**

the effective weight learning mechanism. One can observe that the performance gap is more dramatic with the Holidays dataset. Our interpretation is that the Ukbench dataset has a very balanced category distribution, *i.e.*, exactly 4 images per category, which makes it easier for algorithms to fit such a distribution.

Comparison with State-of-the-art. In Table X, a comprehensive comparison to various state-the-of-art algorithms is presented.

The selected methods can be coarsely divided into two kinds. As the focus of this paper, the first kind aims at feature fusion or diffusion process, including Locally Constrained Mixed Diffusion (LCMD) [39], Contextual Dissimilarity Measure (CDM) [33], kNN Re-ranking [65], and Hello Neighbor [66]. LCMD can be deemed as a kind of naive early fusion, which partly fuses the input similarities into one and propagates on the resulted locally dense data space. It further advocates the use of self-adaptive neighborhoods to automatically determine an appropriate size of the local context in the diffusion process. Yang *et al.* [41] propose a data-driven approach to estimate weights of different similarities, and report mAP 88.3 on the Holidays dataset and N-S score 3.86 on the Ukbench dataset. By using different input similarities, Graph Fusion [2] originally achieves mAP 84.64, and QALF [25] achieves mAP 88.0 on the Holidays dataset.

The second kind facilitates using deep learning for image retrieval, including Gordo *et al.* [62], Convolutional Kernel Network [10], MOP-CNN [64], SPoC [9] and Neural codes [61]. Since this kind of algorithms usually ignores the geometry structure parameterized by one or more similarities, it can be expected that they are compatible with RED for the sake of better retrieval performances.

Moreover, PGM [35] proposes to use spatial verification, and reports the highest mAP 89.2 on the Holidays dataset to our best knowledge. Possibly benefiting from the usage of local descriptors, SN [26] originally achieves N-S score 3.98 on the Ukbench dataset. By contrast, RED reports a very competitive performance, *i.e.*, the best mAP 93.3 on the Holidays dataset and the second best N-S score 3.94 on the Ukbench dataset.

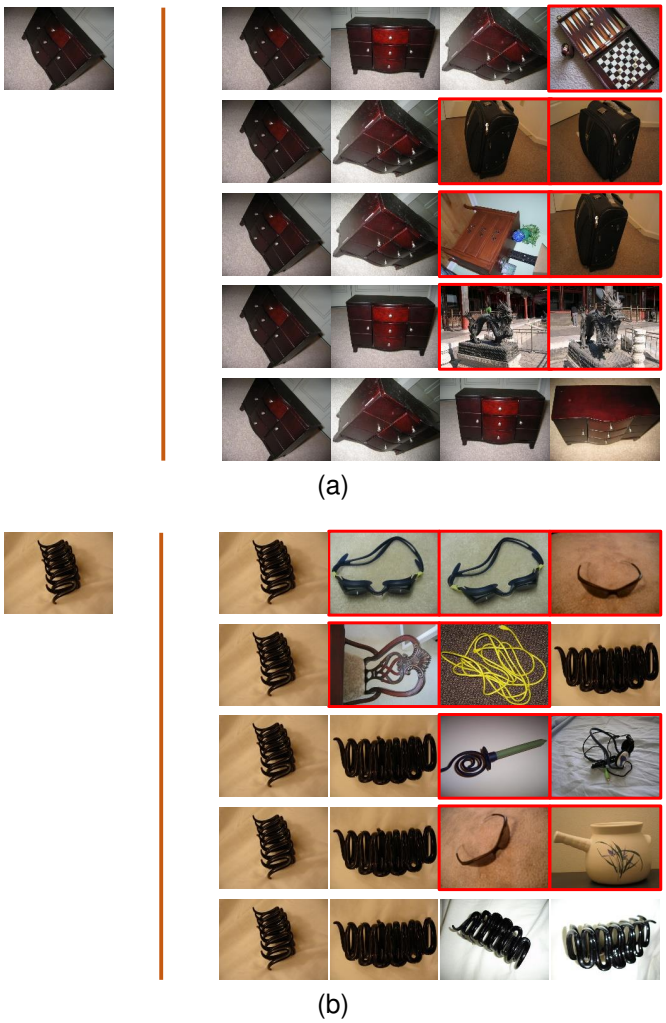


Fig. 4. The sample retrieval results on the Ukbench dataset, where the queries (shown in left) and their top-4 retrieval candidates are given. False positives are in red boxes. The retrieval results of the four baseline similarities, including NetVLAD, SPoC, ResNet and HSV, are presented in the first four rows. The retrieval results of RED are presented in the fifth row.

Qualitative Evaluation. In Fig. 4, we exhibit two indexing results for two image queries on the Ukbench dataset, and find that RED can effectively improve the baseline performances.

C. Discussion

In this section, discussions are primarily done with the Holidays dataset.

Robustness to Noise. Most similarities used in our previous experiments are informative in a sense. To simulate the situation where less informative similarities exist, we manually generate 5 similarities by assigning each pair of objects a random value in the interval $(0, \sqrt{2})$ as their pairwise distance. The image retrieval performance of the 5 similarities is around mAP 0.40. We add the noisy similarities to the four baseline methods for image retrieval (Sec. IV-B), and plot the retrieval

TABLE X
THE COMPARISON WITH THE STATE-OF-THE-ART ON THE HOLIDAYS AND UKBENCH DATASETS.

Datasets	RED	[35]	[26]	[62]	[41]	[63]	[25]	[2]	[9]	[64]	[10]	[61]	[65]	[39]	[33]	[66]
Holidays	**93.3**	89.2	-	89.1	88.3	88.1	88.0	84.6	80.2	80.2	79.3	79.3	76.2	-	-	-
Ukbench	3.94	-	**3.98**	-	3.86	-	3.84	3.83	3.65	-	3.76	3.56	3.52	3.70	3.68	3.67

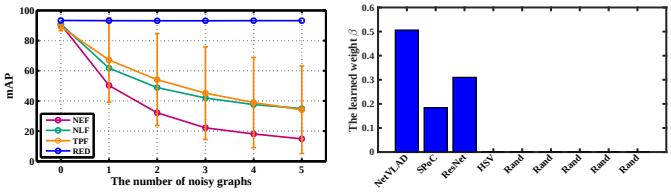


Fig. 5. The retrieval performances with an increasing number of noisy similarities (a), and the learned weights β (b) when 5 noisy similarities are fused.

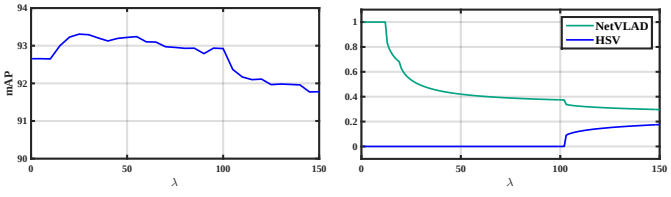


Fig. 6. The retrieval performance (a) and the learned weights β (b) when varying λ.

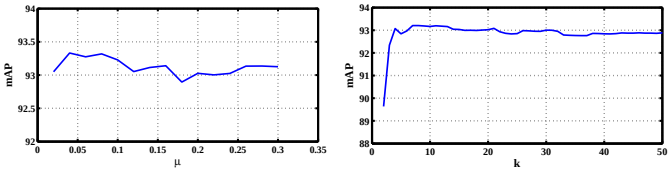


Fig. 7. The influence of μ and the number of nearest neighbors k.

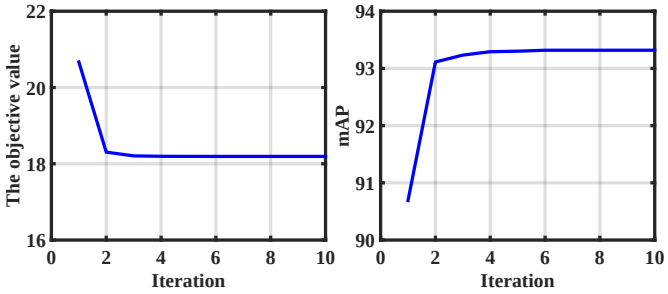


Fig. 8. The objective value and mAP (%) of RED at each iteration.

performances of different fusion with diffusion methods by varying the number of fused noisy similarities in Fig. 5a. For illustration, the performance of tensor product fusion is given by the average value and the standard deviation of mAPs of all combinations of similarity pairs. For naive early fusion and naive early fusion, we only present the results of sum rule, which is demonstrated to be better than product rule in previous experiments.

It can be seen clearly that the performance of RED remains almost unchanged even if 5 noisy similarities are integrated. The reason is that the weights of those 5 similarities learned by RED are all zero, as shown in Fig. 5b. In contrast, naive early fusion, naive late fusion and tensor product fusion encounter a sharp decrease in performance. When 5 noisy similarities are fused, naive early fusion only achieves mAP 14.82. Hence, one can clearly observe the significance of the weight learning part used in RED.

Sensitivity to Parameters. The most important parameter involved in RED is the weight regularizer λ. Fig. 6a shows that the retrieval performances of RED are not so sensitive to the parameter λ. In Fig. 6b, the learned weights for NetVLAD and HSV are illustrated. Firstly, we can observe that RED can tolerate the change of λ, as the curve changes gently. Secondly, when $\lambda \leq 12$, NetVLAD always has weight 1. When $\lambda \leq 100$, HSV has weight 0. It reveals that in a wide range of λ, RED can preserve the discriminative power of informative similarities, and eliminate the negative influence of non-informative similarities. At last, we can conclude that when $\lambda \to \infty$, equal weights will be obtained. It also reveals that the determination of λ relies on the degree of complementary nature among different similarities. If rich complementarity exists, a relatively larger λ is needed.

The other parameters of RED, including k and μ also occur in the conventional diffusion process. In Fig 7, we briefly review their influences on the retrieval performance. As can be seen from Fig. 7a, RED is not sensitive to the change of μ as long as it is in a reasonable range. In Fig. 7b, it is observed that when $k \geq 5$, the performance is significantly improved (around 93 in mAP). When k keeps increasing, the performance drops slightly due to the inclusion of noisy edges on the affinity graph.

Convergence Speed. The objective value of Eq. (17) and the retrieval performance of RED as the iteration increases are given in Fig. 8. As can be seen, RED converges very fast within less than 4 iterations.

Indexing Time Comparison. Table XI presents the comparison of indexing time of different fusion with diffusion methods. As expected, NEF is the most efficient one. Although the time complexity of those four compared methods is in the same scale based on the analysis in Sec. III-E, TPF and RED are generally more time-consuming since the costly similarity propagation step should be run for each input repeatedly.

As for the larger scale retrieval scenario (*e.g.*, the ShapeNet Core55 competition [67] and Oxford105K dataset), RED can be easily adapted by running the similarity learning on the top ranked candidates in the ranking list. Interested readers can refer to other representative postprecessing techniques, such as query expansion [36] and regional diffusion [68], for more

TABLE XI
THE COMPARISON OF INDEXING TIME (SECONDS) OF DIFFERENT FUSION WITH DIFFUSION METHODS.

Datasets	NEF	NLF	TPF	RED
ModelNet40	**1.49**	2.51	23.99	7.86
Holidays	**2.14**	5.26	21.96	35.02

details about this approximated solution.

V. CONCLUSION

In this paper, we focus on the similarity fusion in the framework of diffusion process for robust 3D shape and image retrieval. We have presented a thorough review on several representative algorithms, which are later formally reformulated on the same basis to ensure a fair comparison. Observing that most existing works are sensitive to noisy similarities, we propose Regularized Ensemble Diffusion (RED), with weights positively related to the smoothness of the (tensor product) graph-based manifolds. Comprehensive experiments are conducted with 3D shape retrieval on the ModelNet40 and ModelNet10 datasets, and with image retrieval on the Holidays and Ukbench datasets. The experimental results demonstrate the proposed RED not only outperforms those algorithms focusing on feature fusion or similarity diffusion, but also sets new state-of-the-art performances on the four authoritative retrieval benchmarks.

Although the iterative solver of RED significantly reduces its time complexity, it still requires $O(N^3)$ to finish the similarity propagation step as the conventional diffusion process [16]. Therefore, how to reduce the time complexity of diffusion process [69] to meet the requirement of real-time retrieval can be investigated further. Moreover, RED needs a strategy to efficiently handle out-of-dataset queries. The iteration of diffusion process has to be done each time when a new query point is added. A recently-proposed algorithm called Regional Diffusion [68] have properly addressed this problem, which conducts diffusion on the patch level rather than the object level considered in this paper. Its key concept is to derive a faster iteration-based solver using the conjugate gradient method. Although it cannot be directly applied in our specific scenario, it motivates us to design approximate solutions for the tensor-order iteration on multiple similarities. We leave these issues as our future work.

REFERENCES

[1] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *IJCV*, vol. 60, no. 2, pp. 91–110, 2004.

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

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

[4] B. Ramesh, C. Xiang, and T. H. Lee, "Shape classification using invariant features and contextual information in the bag-of-words model," *Pattern Recognition*, vol. 48, no. 3, pp. 894–906, 2015.

[5] H. Tabia, M. Daoudi, J.-P. Vandeborre, and O. Colot, "A new 3d-matching method of nonrigid and partially similar models using curve analysis," *TPAMI*, vol. 33, no. 4, pp. 852–858, 2011.

[6] H. Tabia, H. Laga, D. Picard, and P.-H. Gosselin, "Covariance descriptors for 3d shape matching and retrieval," in *CVPR*, 2014, pp. 4185–4192.

[7] H. Tabia and H. Laga, "Covariance-based descriptors for efficient 3d shape matching, retrieval, and classification," *TMM*, vol. 17, no. 9, pp. 1591–1603, 2015.

[8] P. Papadakis, I. Pratikakis, T. Theoharis, and S. J. Perantonis, "Panorama: A 3d shape descriptor based on panoramic views for unsupervised 3d object retrieval," *IJCV*, vol. 89, no. 2-3, pp. 177–192, 2010.

[9] A. Babenko and V. Lempitsky, "Aggregating local deep features for image retrieval," in *ICCV*, 2015, pp. 1269–1277.

[10] M. Paulin, M. Douze, Z. Harchaoui, J. Mairal, F. Perronin, and C. Schmid, "Local convolutional features with unsupervised training for image retrieval," in *ICCV*, 2015, pp. 91–99.

[11] Y. Fang, J. Xie, G. Dai, M. Wang, F. Zhu, T. Xu, and E. Wong, "3d deep shape descriptor," in *CVPR*, 2015, pp. 2319–2328.

[12] J. Xie, G. Dai, F. Zhu, E. K. Wong, and Y. Fang, "Deepshape: Deep-learned shape descriptor for 3d shape retrieval," *TPAMI*, vol. 39, no. 7, pp. 1335–1345, 2017.

[13] J. Xie, Y. Fang, F. Zhu, and E. Wong, "Deepshape: Deep learned shape descriptor for 3d shape matching and retrieval," in *CVPR*, 2015, pp. 1275–1283.

[14] Q. Ke and Y. Li, "Is rotation a nuisance in shape recognition?" in *CVPR*, 2014, pp. 4146–4153.

[15] J. Xie, G. Dai, and Y. Fang, "Deep multi-metric learning for shape-based 3d model retrieval," *TMM*, 2017.

[16] M. Donoser and H. Bischof, "Diffusion processes for retrieval revisited," in *CVPR*, 2013, pp. 1320–1327.

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

[18] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "Netvlad: Cnn architecture for weakly supervised place recognition," in *CVPR*, 2016.

[19] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.

[20] S. Zhang, J. Huang, H. Li, and D. N. Metaxas, "Automatic image annotation and retrieval using group sparsity," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 42, no. 3, pp. 838–849, 2012.

[21] L. Zheng, S. Wang, Z. Liu, and Q. Tian, "Packing and padding: Coupled multi-index for accurate image retrieval," in *CVPR*, 2014, pp. 1939–1946.

[22] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3d shapenets: A deep representation for volumetric shape modeling," in *CVPR*, 2015.

[23] H. Jegou, M. Douze, and C. Schmid, "Hamming embedding and weak geometric consistency for large scale image search," in *ECCV*, 2008, pp. 304–317.

[24] S. Bai, Z. Zhou, J. Wang, X. Bai, L. J. Latecki, and Q. Tian, "Ensemble diffusion for retrieval," in *ICCV*, 2017.

[25] L. Zheng, S. Wang, L. Tian, F. He, Z. Liu, and Q. Tian, "Query-adaptive late fusion for image search and person re-identification," in *CVPR*, 2015, pp. 1741–1750.

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

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

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

[29] D. Zhou, J. Weston, A. Gretton, O. Bousquet, and B. Schölkopf, "Ranking on data manifolds," in *NIPS*, 2004, pp. 169–176.

[30] X. Bai, X. Yang, L. J. Latecki, W. Liu, and Z. Tu, "Learning context-sensitive shape similarity by graph transduction," *TPAMI*, vol. 32, no. 5, pp. 861–874, 2010.

[31] B. Wang and Z. Tu, "Affinity learning via self-diffusion for image segmentation and clustering," in *CVPR*, 2012, pp. 2312–2319.

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

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

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

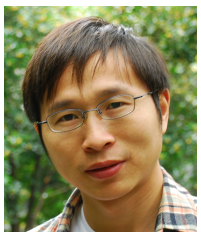
[35] X. Li, M. Larson, and A. Hanjalic, "Pairwise geometric matching for large-scale object retrieval," in *CVPR*, 2015, pp. 5153–5161.

[36] R. Arandjelović and A. Zisserman, "Three things everyone should know to improve object retrieval," in *CVPR*, 2012, pp. 2911–2918.

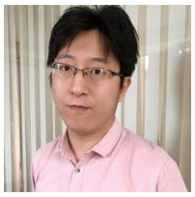
[37] S. Bai, X. Bai, Q. Tian, and L. J. Latecki, "Regularized diffusion process on bidirectional context for object retrieval," *TPAMI*, 2018.
[38] S. Bai, X. Bai, Q. Tian, and L. J. Latecki, "Regularized diffusion process for visual retrieval," in *AAAI*, 2017, pp. 3967–3973.
[39] L. Luo, C. Shen, C. Zhang, and A. van den Hengel, "Shape similarity analysis by self-tuning locally constrained mixed-diffusion," *TMM*, vol. 15, no. 5, pp. 1174–1183, 2013.
[40] S. Zhang, M. Yang, T. Cour, K. Yu, and D. N. Metaxas, "Query specific fusion for image retrieval," in *ECCV*, 2012, pp. 660–673.
[41] F. Yang, B. Matei, and L. S. Davis, "Re-ranking by multi-feature fusion with diffusion for image retrieval," in *WACV*, 2015, pp. 572–579.
[42] D. C. G. Pedronette and R. d. S. Torres, "Rank diffusion for context-based image retrieval," in *ICMR*, 2016, pp. 321–325.
[43] D. C. G. Pedronette and R. D. S. Torres, "Image re-ranking and rank aggregation based on similarity of ranked lists," *Pattern Recognition*, vol. 46, no. 8, pp. 2350–2360, 2013.
[44] D. C. G. Pedronette, O. A. Penatti, and R. d. S. Torres, "Unsupervised manifold learning using reciprocal knn graphs in image re-ranking and rank aggregation tasks," *Image and Vision Computing*, vol. 32, no. 2, pp. 120–130, 2014.
[45] Y. Zhou, X. Bai, W. Liu, and L. J. Latecki, "Fusion with diffusion for robust visual tracking," in *NIPS*, 2012, pp. 2978–2986.
[46] Y. Zhou, X. Bai, W. Liu, and L. J. Latecki, "Similarity fusion for visual tracking," *IJCV*, pp. 1–27, 2016.
[47] S. Bai, S. Sun, X. Bai, Z. Zhang, and Q. Tian, "Improving context-sensitive similarity via smooth neighborhood for object retrieval," *Pattern Recognition*, 2018.
[48] H. Su, S. Maji, E. Kalogerakis, and E. G. Learned-Miller, "Multi-view convolutional neural networks for 3d shape recognition," in *ICCV*, 2015, pp. 945–953.
[49] S. Bai, X. Bai, Z. Zhou, Z. Zhang, and L. J. Latecki, "Gift: A real-time and scalable 3d shape search engine," in *CVPR*, 2016.
[50] C. R. Qi, H. Su, M. Niessner, A. Dai, M. Yan, and L. J. Guibas, "Volumetric and multi-view cnns for object classification on 3d data," in *CVPR*, 2016.
[51] J. Delhumeau, P.-H. Gosselin, H. Jégou, and P. Pérez, "Revisiting the vlad image representation," in *ACM international conference on Multimedia*, 2013, pp. 653–656.
[52] P. Papadakis, I. Pratikakis, S. Perantonis, and T. Theoharis, "Efficient 3d shape matching and retrieval using a concrete radialized spherical projection representation," *Pattern Recognition*, vol. 40, no. 9, pp. 2437–2452, 2007.
[53] E. Johns, S. Leutenegger, and A. J. Davison, "Pairwise decomposition of image sequences for active multi-view recognition," in *CVPR*, 2016.
[54] J. Xie, M. Wang, and Y. Fang, "Learned binary spectral shape descriptor for 3d shape correspondence," in *CVPR*, 2016, pp. 3309–3317.
[55] C. Wang, M. Pelillo, and K. Siddiqi, "Dominant set clustering and pooling for multi-view 3d object recognition," in *BMVC*, 2017.
[56] M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz, "Rotation invariant spherical harmonic representation of 3d shape descriptors," in *SGP*, 2003, pp. 156–164.
[57] D. Y. Chen, X. P. Tian, Y. T. Shen, and M. Ouhyoung, "On visual similarity based 3d model retrieval," *Comput. Graph. Forum*, vol. 22, no. 3, pp. 223–232, 2003.
[58] B. Shi, S. Bai, Z. Zhou, and X. Bai, "Deeppano: Deep panoramic representation for 3-d shape recognition," *IEEE Signal Processing Letters*, vol. 22, no. 12, pp. 2339–2343, 2015.
[59] T. Furuya and R. Ohbuchi, "Deep aggregation of local 3d geometric features for 3d model retrieval," in *BMVC*, 2016.
[60] H. Jegou, F. Perronnin, M. Douze, J. Sánchez, P. Perez, and C. Schmid, "Aggregating local image descriptors into compact codes," *TPAMI*, vol. 34, no. 9, pp. 1704–1716, 2012.
[61] A. Babenko, A. Slesarev, A. Chigorin, and V. Lempitsky, "Neural codes for image retrieval," in *ECCV*, 2014, pp. 584–599.
[62] A. Gordo, J. Almazán, J. Revaud, and D. Larlus, "Deep image retrieval: Learning global representations for image search," in *ECCV*, 2016, pp. 241–257.
[63] M. Shi, Y. Avrithis, and H. Jégou, "Early burst detection for memory-efficient image retrieval," in *CVPR*, 2015, pp. 605–613.
[64] Y. Gong, L. Wang, R. Guo, and S. Lazebnik, "Multi-scale orderless pooling of deep convolutional activation features," in *ECCV*, 2014, pp. 392–407.
[65] X. Shen, Z. Lin, J. Brandt, S. Avidan, and Y. Wu, "Object retrieval and localization with spatially-constrained similarity measure and k-nn re-ranking," in *CVPR*, 2012, pp. 3013–3020.
[66] D. Qin, S. Gammeter, L. Bossard, T. Quack, and L. Van Gool, "Hello neighbor: Accurate object retrieval with k-reciprocal nearest neighbors," in *CVPR*, 2011, pp. 777–784.
[67] M. Savva, F. Yu, H. Su, M. Aono, B. Chen, D. Cohen-Or, W. Deng, H. Su, S. Bai, X. Bai *et al.*, "Shrec16 track large-scale 3d shape retrieval from shapenet core55," in *3DOR*, 2016.
[68] A. Iscen, G. Tolias, Y. Avrithis, T. Furon, and O. Chum, "Efficient diffusion on region manifolds: Recovering small objects with compact cnn representations," in *CVPR*, 2017.
[69] H. Tong, C. Faloutsos, and J. Pan, "Fast random walk with restart and its applications," in *ICDM*, 2006, pp. 613–622.

Song Bai received the B.S. and Ph.D. degree in Electronics and Information Engineering from Huazhong University of Science and Technology (HUST), Wuhan, China in 2013 and 2018, respectively. His research interests include image retrieval and classification, 3D shape recognition, person re-identification, semantic segmentation and deep learning. More information can be found in his homepage: http://songbai.site/.

Zhichao Zhou received the B.S. degree in Electronics and Information Engineering from Huazhong University of Science and Technology (HUST), Wuhan, China in 2015. He is currently working toward the M.S. degree in the School of Electronic Information and Communications, HUST. His research interests include shape analysis, deep learning and its applications.



Jingdong Wang is a Senior Researcher with the Visual Computing Group, Microsoft Research Asia. He received the B.Eng. and M.Eng. degrees from the Department of Automation, Tsinghua University, Beijing, China, in 2001 and 2004, respectively, and the PhD degree from the Department of Computer Science and Engineering, the Hong Kong University of Science and Technology, Hong Kong, in 2007. His areas of interest include deep learning, large-scale indexing, human understanding, and person re-identification. He is an Associate Editor of IEEE TMM and IEEE TCSVT, and is an area chair (or SPC) of some prestigious conferences, such as CVPR, ICCV, ECCV, ACM MM, IJCAI, and AAAI. He is a Fellow of IAPR.



Xiang Bai received the B.S., M.S. and Ph.D. degrees from 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 National Center of Anti-Counterfeiting Technology, HUST. His research interests include object recognition, shape analysis, scene text recognition and intelligent systems. He is a senior member of IEEE.

Longin Jan Latecki is a full professor of computer science at Temple University in Philadelphia. His main research interests include computer vision and pattern recognition. He has published over 240 research papers and books. He is an editorial board member of Pattern Recognition and Computer Vision and Image Understanding. He received the annual Pattern Recognition Society Award together with Azriel Rosenfeld for the best article published in the journal Pattern Recognition in 1998. His research has been funded by NSF, AFOSR, Los Alamos National Lab, DOE, NIST, and Johnson and Johnson Pharmaceutical Research.

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 2008-2012 and a tenure-track Assistant Professor from 2002-2008. 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.