Affinity Learning with Diffusion on Tensor Product Graph

Xingwei Yang, Lakshman Prasad, and Longin Jan Latecki, *Senior Member, IEEE*

Abstract—In many applications, we are given a finite set of data points sampled from a data manifold and represented as a graph with edge weights determined by pairwise similarities of the samples. Often the pairwise similarities (which are also called affinities) are unreliable due to noise or due to intrinsic difficulties in estimating similarity values of the samples. As observed in several recent approaches, more reliable similarities can be obtained if the original similarities are diffused in the context of other data points, where the context of each point is a set of points most similar to it.

Compared to the existing methods, our approach differs in two main aspects. First, instead of diffusing the similarity information on the original graph, we propose to utilize the tensor product graph (TPG) obtained by the tensor product of the original graph with itself. Since TPG takes into account higher order information, it is not a surprise that we obtain more reliable similarities. However, it comes at the price of higher order computational complexity and storage requirement. The key contribution of the proposed approach is that the information propagation on TPG can be computed with the same computational complexity and the same amount of storage as the propagation on the original graph. We prove that a graph diffusion process on TPG is equivalent to a novel iterative algorithm on the original graph, which is guaranteed to converge. After its convergence we obtain new edge weights that can be interpreted as new, learnt affinities. We stress that the affinities are learned in unsupervised setting.

We illustrate the benefits of the proposed approach for data manifolds composed of shapes, images, and image patches on two very different tasks of image retrieval and image segmentation. With learned affinities, we achieve the bull's eye retrieval score of **99.99%** on MPEG-7 shape dataset, which is much higher than the state-of-the-art algorithms. When the data points are image patches, the NCut with the learned affinities not only significantly outperforms the NCut with the original affinities, but it also outperforms state-of-the-art image segmentation methods.

Index Terms—Diffusion Process, Tensor Product Graph, Affinity Learning, Image Retrieval, Image Segmentation.

1 Introduction

One of the most important properties of the proposed tensor product graph (TPG) diffusion is the fact that it able to capture the geometry of the data manifold. Usually, the dimensionality of the feature space (number of features) is large, but the intrinsic dimensionality of the data manifold is small. At the same time the shape of the data manifold is very important, and it is often curved. Formally this means that it is not a convex subset of the feature space. We observe that only if the data manifold is convex, the Euclidean distances are an adequate measure of distances between the data points. Thus, if the data manifold is curved, the Euclidean distances are inadequate. The proposed TPG diffusion is able to robustly estimate (or learn) distances between data points that correspond to geodesic distances on the data manifold. We illustrate this fact on a toy dataset in Fig. 1. We compare the classification performance of TPG diffusion to spectral embedding [1], which is a standard dimensionality reduction method. The goal of dimensionality reduction methods is to capture the geometry of the data manifold by embedding it into a lower dimensional space with the hope that the Euclidean distances in the new space approximate the geodesic distances on the data manifold. The spectral embedding computes the embedding based on eigenvalue decomposition of the data similarity matrix. In contrast, the TPG diffusion estimates the geodesic distances directly on the data manifold.

In each half moon in Fig. 1, one point is selected as a labeled point (marked with a star). We use the simplest possible classifier, i.e., 1NN classifier. It assigns an (unlabeled) data point to the closest labeled point (one of the two stars). The ground truth of each point is known beforehand (marked with red and blue colors). Table 1 shows the classification rates for different noise levels for the Euclidean distance, TPG diffusion, and spectral embedding. Even with no noise the Euclidean distance can only classify correctly 76% of the data points. This shows that the geodesic distances on this data manifold are not well approximated with the Euclidean distances. The classification rate of 100% of the TPG diffusion clearly demonstrates that it has no difficulty to learn the geodesic distances. In contrast spectral embedding can only achieve 84.83%. (For spectral embedding, we use the two most important eigenvectors of the similarity matrix to embed the data into 2D.) With

- *X. Yang and L. J. Latecki are with Department of Computer and Information Sciences, Temple University, Philadelphia, USA. E-mails: xingwei@temple.edu, latecki@temple.edu*
- *L. Prasad is with LANL, Los Alamos, USA. E-mail: prasad@lanl.gov*

Digital Object Indentifier 10.1109/TPAMI.2012.60

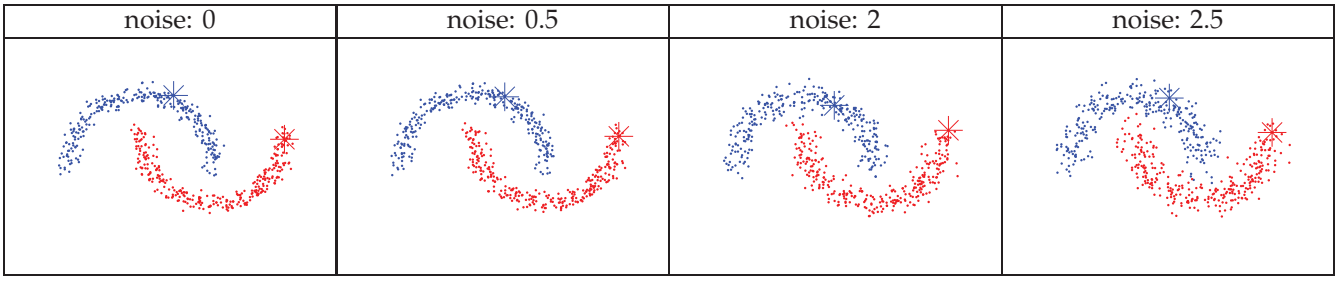

Fig. 1. Toy example dataset with added Gaussian noise. Two labeled data points are marked with stars.

TABLE 1
The 1NN classification rates with respect to two labeled data points on the dataset in Fig. 1.

noise level	0	0.5	1	1.5	2	2.5
Euclidean Distance	76%	75.93%	75.73%	75.65%	75.82%	74.97%
TPG Diffusion	100%	100%	97.28%	85.03%	79.05%	77.72%
Spectral Embedding	84.83%	84.17%	82.88%	80.86%	76.19%	73.83%

Fig. 2. Three example shapes form each of the three shape classes from the MPEG-7 shape dataset.

the increased amount of noise performance of both spectral embedding and TPG diffusion drop, but TPG diffusion still outperforms spectral embedding. Of course, if the noise significantly corrupts the manifold structure, as is the case for noise level of 2.5, none of the methods can do much better than the baseline Euclidean distance.

The noise is generated by displacing all data points with added Gaussian noise. The level of noise is the ratio between the Gaussian sigma and the average distance of each point to its nearest neighbor in the whole dataset. The classification results reported in Fig. 1 are the mean results of 10 runs of the experiment.

In order to further motivate the proposed approach, we consider the popular MPEG-7 shape dataset containing binary images of 1400 shapes. The dataset is divided into 70 classes with 20 shapes in each class. In Fig. 2 we show three shapes from three classes: Device1, Device2, and Device9. We observe that some shapes from Device1 and Device2 are more similar to each other than to other shapes in the same class, which illustrates one of the main problems of shape based object recognition. Since shapes from the same class often exhibit large intra class variance, a shape similarity measure must tolerate this variance, but at the same time it must be discriminative enough to differentiate shapes from different classes. It seems to be impossible to solve this problem if two shapes are only compared in isolation form other shapes.

A solution to this problem has been pointed out in a sequence of recent papers, [2], [3], [4], [5], where the pairwise comparisons are augmented by the context of other shapes. As demonstrated by these papers considering the data manifold structure defined by other shapes significantly improves the performance of ranking/retrieval. The basic idea is inspired by the success of google PageRank ranking. The data manifold is represented as a graph with edge weights determined by the initial pairwise similarity values. Then the pairwise similarities between the query and each database object are reevaluated in the context of other database objects, where the context of each object is a set of other objects most similar to it and the reevaluation is obtained by propagating the similarity information following structure of the weighted edge links in the graph. The reevaluation is closely related to random walks on the graph, e.g., [6], [7].

One of the standard ways of propagating the similarity information is by a diffusion process on a weighted graph [6], [8]. In Fig. 3(a), we show part of the MPEG-7 affinity matrix A representing the three shape classes: Device1, Device2, and Device9. The affinities are computed based on pairwise shape comparison with aspect shape context (ASC) [9]. Fig. 3(b) show the affinities after the classic diffusion on the MPEG-7 graph represented by A. While the original affinity matrix in (a) is sparse in that some shapes in the same class do not exhibit high similarities, the diffusion is able to propagate the similarity information and relate objects in the same classes. However, in doing so, the distinction between classes Device1 and Device2 has been lost. In contrast, the proposed TPG diffusion is able to propagate the within class similarities and still preserves the between class separation as shown in Fig. 3(c). This demonstrates that the diffusion on TPG is able to better propagate the inner class affinities while at the same time preserving the intra class differences.

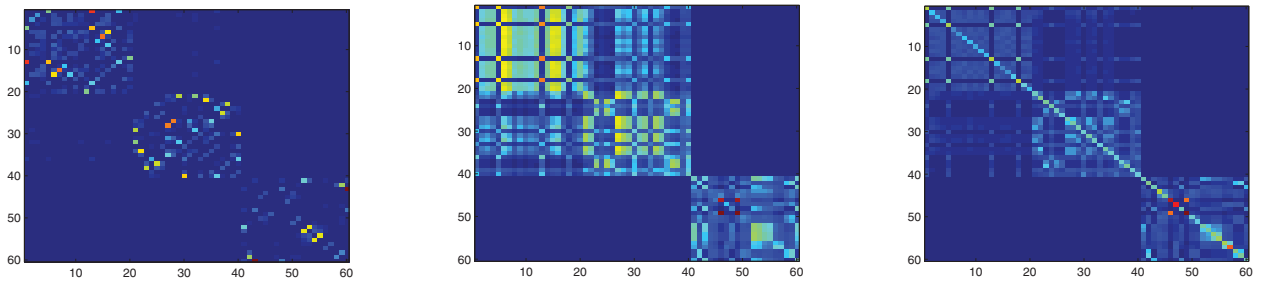

Fig. 3. (a) The original affinities of 60 shapes belonging to 3 different shape classes of MPEG-7 shape dataset. (b) The affinities learned with the classic diffusion on the original graph. (c) The affinities learned with the proposed TPG diffusion. While the structure of 3 clusters is clearly visible in (c), the first two shape classes form a single cluster in (b). (The self similarities in (a) are set to zero for better visualization.)

Compared to the existing methods, our approach differs in two main aspects. First, instead of propagating the similarity information on the original graph, we propose to utilize the tensor product graph (TPG) obtained by the tensor product of the original graph with itself. Since TPG takes into account a higher order information compared to the original methods, it comes at no surprise that we obtain better retrieval performance. Higher order information has been utilized in many applications before, e.g., [10], [11], but it comes at the price of higher order computational complexity and storage requirement. The key feature of the proposed approach is that the information propagation on TPG can be computed with the same computational complexity and the same amount of storage as the propagation on the original graph. We utilize a graph diffusion process to propagate the similarity information on TPG, but we never compute it directly on TPG. Instead, we derive a novel iterative algorithm to compute it directly on the original graph, which is guaranteed to converge. After its convergence we obtain new edge weights that can be interpreted as new, learnt similarities. They are then used for final retrieval ranking.

Fig. 4 compares the retrieval results of the learnt similarities to those of original similarities. The queries are shown in the first column. The first row shows retrieval results of an original shape similarity measure ASC [9] on the MPEG-7 dataset. The results after the proposed diffusion on TPG are shown in the second row. As can be seen the proposed similarity learning with diffusion on TPG is able to correct wrong retrieval results of the original similarities.

We demonstrate the benefits of learned affinities on two very different applications. First in Section 4 we focus on shape and image retrieval, and demonstrate that the learned similarities lead to better retrieval results. This demonstrates that the distances corresponding to the learned similarities better estimate the true geodesic distances on the data manifold. Besides shape retrieval, we also show that learnt similarities lead to improved clustering results. As a special and particularly challenging case we study image segmentation. Popular, unsupervised segmentation algorithm, such as Normalized Cut (NCut) [12] and Mean Shift [13] can be viewed as clustering methods. In Section 5 we demonstrate that the NCut image segmentation algorithm with the learned affinities not only significantly outperforms the NCut with the original affinities, but it also outperforms state-of-the-art image segmentation methods. The benefits of learned similarities are not limited to the two applications. For example, K nearest neighbor (KNN) classifiers perform better when distances are closer to the intrinsic distances on the data manifold.

We present the related work in the next section. The construction of TPG and the proposed approach to learning affinities by diffusion on TPG are introduced in Section 3. The diffusion process on TPG is described in Section 3.1 and the proposed lower complexity iterative algorithm to compute it in Section 3.2, where their equivalence is also proved.

2 Related Work

Information propagation on a weighed graph representing a given dataset has been considered in the context of image retrieval. Zhou et al. [2] proposed to improve ranking of retrieved objects by utilizing the data manifold structure. In other words, the context information of similar database objects is used to improve ranking results. Recently, Bai et al. [5] proposed to utilize the context information for shape retrieval with Label Propagation, which was originally designed for semi-supervised learning. Kontschieder et al. [4] proposed a different way to utilize the context information to improve the performance. A graph diffusion process is utilized for retrieval in Yang et al. [3], where a Locally Constraint Diffusion Process (LCDP) is proposed. LCDP has been used in [9] and [14] to improve their results. Graph diffusion has also

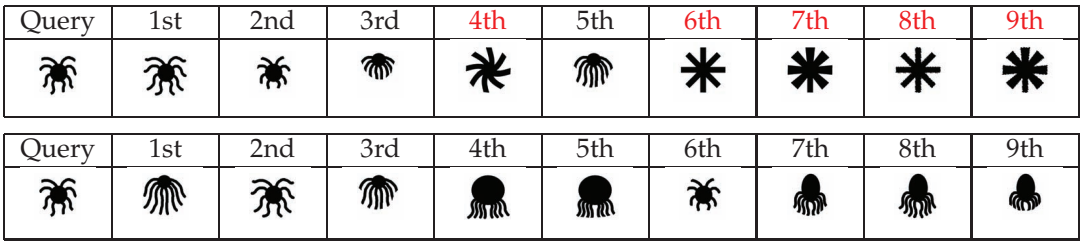

Fig. 4. First row: the query and the retrieval results with the original ASC shape similarity measure on the MPEG-7 shape dataset. Second row: the same query and the retrieval results with similarities learned by the proposed diffusion on TPG.

been utilized for manifold embedding [8]. However, the distances relation among data after embedding are not suitable for retrieval, which is demonstrated in our experimental results. Furthermore, Bai et al. [15] introduce a novel retrieval methods, co-transduction, motivated by co-training. It fuses different similarity/dissimilarity measures to better determine the relation between objects.

Content based image retrieval is getting more attention recently. The classic methods [16], [17] use the bag of feature image representation for image retrieval. Jegou et al. [18] refine the affinity among the objects by enforcing the neighborhood relation to be symmetric. To reduce the complexity of current algorithms, Jegou et al. [19] proposes a novel features extraction approach. Different from the current methods, we improve the image retrieval results by utilizing the higher order information of the TPG. The iterative algorithm for diffusion process on TPG was presented by the authors in a conference paper [20].

The proposed application of tensor graph diffusion to image segmentation is most closely related to [21]. However, there are two key differences. First, we learn new affinities on a Tensor Product Graph that allows us to utilize higher order relations. This is able to better reveal the intrinsic relation between data. Second, we utilize hierarchical segmentations in our framework. Thus, the neighborhood information can be integrated into the affinity learning procedure. As proved by many other papers [22], [23], [24], [25], [26], this information complements purely local information to a great advantage. We use the finest level of hierarchical image segmentation as the basic unit instead of pixels. The superpixels are not only more informative than pixels [27] but also reduce the time complexity and memory requirement. Once new affinities of image regions are learned, we utilize the classical Normalized Cuts (NCut) algorithm by Shi and Malik [12] to obtain the final segmentation results.

3 Affinity Learning

In this section we describe a novel context-sensitive affinity learning algorithm. It is introduced as a diffusion process on a Tensor Product Graph (TPG). However, the size of TPG is quadratic as compared to the original graph, which makes the diffusion on the TPG impractical on large datasets due to both high computation time and high memory requirement. To solve this problem, we propose a novel iterative algorithm on the original graph (Section 3.2), and prove that it is equivalent to the diffusion process on TPG. Consequently, both time complexity and memory requirements of the iterative algorithm are comparable to other affinity learning methods like diffusion on the original graph [6] or LGC[2], [7].

The most popular representations of data manifolds given by a set of data points is an edge-weighted graph $G = (V, A)$, where $V = \{v_1, ..., v_n\}$ is the set of vertices representing the data points and A is the graph adjacency matrix $A(i, j) = (a_{ij})$ for $i, j = 1, \ldots, n$. The edge weight a_{ij} from v_i to v_j represents a pairwise similarity (or equivalently affinity) between data points v_i to v_j.

It is well known that a graph diffusion process is able to reveal the intrinsic relation between objects [8], [6]. Probably the simplest realization of a diffusion process on a graph is by computing powers of the graph matrix, i.e., the edge weights at time t are given by A^t. Usually, the time is discrete and t corresponds to the iteration number. However, this process is sensitive to the number of iterations [28]. For example, if the sum of each row of A is smaller than one, as we assumed, then it converges to zero matrix, in which case determining a right stopping time t is critical. In order to make the graph diffusion process independent from the number of iteration, accumulation between different numbers of iterations is widely used [28]. Following this strategy, we consider the graph diffusion process defined as

$$A^{(t)} = \sum_{i=0}^{t} A^i \tag{1}$$

We assume that A is nonnegative and the sum of each row is smaller than one. A matrix A that satisfies these requirements can be easily created from a stochastic matrix. The assumption that the sum of each row of $A < 1$ is equivalent to the fact that the the maximum of the row-wise sums of matrix $A < 1$. It is known that the maximum of the absolute values of the eigenvalues is bounded by the the maximum of the row-

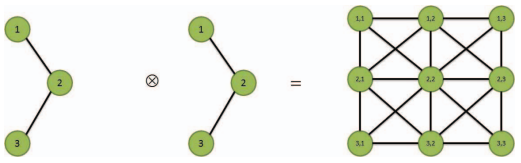

Fig. 5. An example of a tensor product graph. We do not show the self connections (loops), but each node has a loop in $\mathbb{G}$.

wise sums. Therefore, we obtain that the maximum of the absolute values of the eigenvalues of A is smaller than one. Consequently, (1) converges to a fixed and nontrivial solution given by $\lim_{t\to\infty} A^{(t)} = (I - A)^{-1}$, where I is the identify matrix.

3.1 Diffusion Process on Tensor Product Graph

The Tensor Product Graph (TPG) $\mathbb{G} = G \otimes G$ is defined as $\mathbb{G} = (V \times V, \mathbb{A})$. Thus, each vertex of $\mathbb{G}$ is a pair of vertices in G, and consequently, it is indexed with a pair of indices. The adjacency matrix of $\mathbb{G}$ is defined as $\mathbb{A} = A \otimes A$, where $\otimes$ is the Kronecker product [29], [30]. In particular, for $\alpha, \beta, i, j = 1 \ldots, n$ we have

$$\mathbb{A}(\alpha, \beta, i, j) = A(\alpha, \beta) \cdot A(i, j) = a_{\alpha,\beta} \cdot a_{i,j}.$$

Thus, if $A \in \mathbb{R}^{n\times n}$, then $\mathbb{A} = A \otimes A \in \mathbb{R}^{nn\times nn}$. An example is shown in Fig. 5.

We define the **diffusion process on TPG** as

$$\mathbb{A}^{(t)} = \sum_{i=0}^{t} \mathbb{A}^i. \tag{2}$$

Since the edge weights of TPG relate 4 tuples of original vertices, $\mathbb{G}$ contains high order information than the input graph. The higher order information is helpful for revealing the intrinsic relation between objects, which is obtained by the diffusion process on TPG.

As is the case for (1), the process (2) also converges to a fixed and nontrivial solution

$$\lim_{t\to\infty} \mathbb{A}^{(t)} = \lim_{t\to\infty} \sum_{i=0}^{t} \mathbb{A}^i = (I - \mathbb{A})^{-1}. \tag{3}$$

To show this, we only need to show that the sum of each row of $\mathbb{A}$ is smaller than 1, i.e., $\sum_{\beta,j} \mathbb{A}(\alpha\beta, ij) < 1$, where β, j both range from 1 to n. This holds, since

$$\sum_{\beta j} \mathbb{A}(\alpha\beta, ij) = \sum_{\beta j} a_{\alpha\beta} a_{ij} = \sum_{\beta} a_{\alpha\beta} \sum_{j} a_{ij} < 1. \tag{4}$$

Consequently, (3) provides a closed form solution for the diffusion process on TPG. However, our goal was to utilize TPG to learn new affinities on the original graph G. i.e., to obtain a new affinity matrix A^* of size $n \times n$. The matrix A^* containing the **learned affinities** is defined as

$$A^* = \mathrm{vec}^{-1}((I - \mathbb{A})^{-1}\ \mathrm{vec}(I)), \tag{5}$$

where I ia an $n \times n$ identity matrix and vec is an operator that stacks the columns of a matrix one after the next into a column vector. Formally, for a given $m \times n$ matrix B

$$\mathrm{vec}(B) = (b_{11}, \ldots, b_{m1}, b_{12}, \ldots, b_{m2}, \ldots, b_{1n}, \ldots, b_{mn})^T.$$

Since $\mathrm{vec} : \mathbb{R}^{m\times n} \to \mathbb{R}^{mn}$ is an isomorphism, its inverse exists, and we denote it with vec^{-1}.

To summarize, for the input affinity matrix A, the new learned affinities are given by matrix A^* defined in (5). However, the affinity learning with the proposed diffusion process on TPG (2) is impractical for large graphs due to high storage and computing cost. The diffusion on the original graph G requires $O(n^2)$ storage (number of the matrix elements) and its computation cost is determined by the cost of matrix inversion, which is $O(n^3)$ for Gauss-Jordan elimination or about $O(n^{2.4})$ for the Coppersmith-Winograd algorithm. In contrast the diffusion on TPG requires $O(n^4)$ storage and its computation cost is $O(n^6)$ for Gauss-Jordan elimination or about $O(n^{4.8})$ for the Coppersmith-Winograd algorithm. Therefore, we propose a novel iterative algorithm in Section 3.2 to compute (2). Its storage and computation cost is comparable to the diffusion on the original graph, since it is executed on the original graph.

3.2 Iterative Algorithm for Diffusion on TPG

We define $Q^{(1)} = A$ and

$$Q^{(t+1)} = A\ Q^{(t)}\ A^T + I, \tag{6}$$

where I is the identity matrix. We iterate (6) until convergence. Let us denote the limit matrix by $Q^* = \lim_{t\to\infty} Q^{(t)}$. The proof of the convergence of (6) and a closed form expression for Q^* both follow from the following key equation

$$\lim_{t\to\infty} Q^{(t)} = Q^* = A^* = \mathrm{vec}^{-1}((I - \mathbb{A})^{-1}\ \mathrm{vec}(I)). \tag{7}$$

The remainder of this section is devoted to prove this equation. Since $Q^* = A^*$, we obtain that the iterative algorithm on the original graph G defined by (6) yields the same affinities as the TPG diffusion process on $\mathbb{G}$.

In order to prove (7), we first transform (6) to

$$\begin{aligned} Q^{(t+1)} &= A\ Q^{(t)}\ A^T + I = A(A\ Q^{(t-1)}\ A^T + I)A^T + I \\ &= A^2\ Q^{(t-1)}\ (A^T)^2 + A\ I\ A^T + I = \ldots \\ &= A^t\ A\ (A^T)^t + A^{t-1}\ I\ (A^T)^{t-1} + \ldots + I \\ &= A^t\ A\ (A^T)^t + \sum_{i=0}^{t-1} A^i\ I\ (A^T)^i \end{aligned} \tag{8}$$

Since (by our assumption) sum of each row of $A < 1$, we have $\lim_{t\to\infty} A^t\ A\ (A^T)^t = 0$, and consequently,

$$Q^* = \lim_{t\to\infty} Q^{(t+1)} = \lim_{t\to\infty} \sum_{i=0}^{t-1} A^i\ I\ (A^T)^i \tag{9}$$

We observe that the following identity holds

$$\mathrm{vec}(A\ S\ A^T) = (A \otimes A)\mathrm{vec}(S) = \mathbb{A}\ \mathrm{vec}(S), \quad (10)$$

where we recall that $\otimes$ is the Kronecker product. As a consequence we obtain for every $i = 0, 1, 2, \ldots$

$$\mathrm{vec}(A^i\ I\ (A^T)^i) = \mathbb{A}^i \mathrm{vec}(I). \quad (11)$$

Our proof of (11) is by induction. Suppose

$$\mathrm{vec}(A^k\ I\ (A^T)^k) = \mathbb{A}^k \mathrm{vec}(I)$$

holds for $i = k$, then for $i = k + 1$ we have

$$\mathrm{vec}(A^{k+1}\ I\ (A^T)^{k+1}) = \mathrm{vec}(A\ (A^k\ I\ (A^T)^k)\ A^T)$$
$$= \mathbb{A}\ \mathrm{vec}(A^k\ I\ (A^T)^k) = \mathbb{A}\ \mathbb{A}^k \mathrm{vec}(I) = \mathbb{A}^{k+1}\mathrm{vec}(I)$$

From (11) and from the fact that vec of a sum of matrices is sum of their vec's, we obtain

$$\mathrm{vec}(\sum_{i=0}^{t-1}(A)^i\ I\ ((A)^T)^i) = \sum_{i=0}^{t-1}\mathbb{A}^i \mathrm{vec}(I). \quad (12)$$

Finally from (9) and (12), we derive

$$\mathrm{vec}(Q^*) = \lim_{t\to\infty} \mathrm{vec}(\sum_{i=0}^{t-1} A^i\ I\ (A^T)^i) = \lim_{t\to\infty} \sum_{i=0}^{t-1}(\mathbb{A}^i\ \mathrm{vec}(I))$$
$$= (\lim_{t\to\infty} \sum_{i=0}^{t-1} \mathbb{A}^i)\ \mathrm{vec}(I) = (I - \mathbb{A})^{-1}\ \mathrm{vec}(I) \quad (13)$$

This proves our key equation (7). Hence the iterative algorithm (6) on G yields the same affinities as the TPG diffusion process on $\mathbb{G}$.

Since our iterative algorithm works on the original graph G, both its storage and computational cost requirements are significantly lower than those of the TPG diffusion process. It requires $O(n^2)$ storage and its computation cost is determined by the cost of matrix multiplication, which is $O(n^3)$ for direct implementation or about $O(n^{2.4})$ for the Coppersmith-Winograd algorithm. Consequently, if the number of iterations is $t = T$, then its computational cost is $O(Tn^3)$ or $O(Tn^{2.4})$, correspondingly.

Graph G in Fig. 5 provides a simple example to illustrate the fact that the diffusion on the TPG considers the information from more edge weights than the diffusion on the original graph. For simplicity we compare only the second iteration, i.e., we compare $A^{(2)}$ to $Q^{(2)}$ and focus on the edge weight between 1 and 3. Since there is no edges between 1 and 3 in G, we have $a_{13} = a_{31} = 0$. Therefore, in $A^{(2)}$ we have $a_{13}^{(2)} = a_{12} \cdot a_{23}$. The corresponding weight of the edge between 1 and 3 in $Q^{(2)}$ is given by

$$q_{13}^{(2)} = a_{12} \cdot a_{23} \cdot (a_{11} + a_{22}) + a_{12} \cdot a_{33} \cdot a_{33}.$$

While $a_{13}^{(2)}$ only depends on the edge weights a_{12} and a_{23}, $q_{13}^{(2)}$ also depends on the self similarities a_{11}, a_{22}, a_{33}. In particular, we can have $a_{13}^{(2)} < q_{13}^{(2)}$ if $a_{11} + a_{22} > 1$, but we can also have $a_{13}^{(2)} > q_{13}^{(2)}$. Thus, TPG diffusion utilizes more information to determine the strength of the connection between 1 and 3 than just the connections a_{12} and a_{23} considered by the diffusion on the original graph. The difference in the number of connections considered is even more dramatic for $t > 2$. TPG diffusion also utilizes the self-reinforcement in that the strength of the connections depends on the ratio between the similarity of each database object to itself and the sum of its similarities to other objects.

4 Performance Evaluation with Image Retrieval Rates

To demonstrate the advantages of the learned distances, we show that they lead to significant improvement of data manifold distances, which we measure using shape and image retrieval rates. We emphasize that the proposed approach is not a stand alone image retrieval method. It is an affinity (or distance) learning algorithm. In Section 5 we demonstrate that learned affinities also improve image segmentation results. Our goal is to demonstrate that the learned affinities yield better results than the original affinities. This implies that the learned affinities better estimate the true geodesic distances on the underlying data manifolds. In particular, this means that the learned affinities can improve any distance based image retrieval method, e.g., [31]. On all test datasets, the proposed method achieves excellent results, which are significantly better than baseline results obtained with original affinities.

Since our iterative algorithm to compute the TPD diffusion is guaranteed to converge, we only need to ensure that the number of iterations is not too small. It is set to 200 for all test datasets.

If pairwise distances are provided for a given dataset, we transform the distances to similarities with the method introduced in [32]. It applies a Gaussian to distances with variable sigma, which is determined by distances to neighbor data points. We denote the so obtained a similarity matrix with W. Then we row wise normalize W to a stochastic matrix so that sum of each row is one. Finally, we obtain the input affinity matrix A by $a_{ij} = w_{ij}$ if $j \in kDN(i)$ and $a_{ij} = 0$ otherwise, where kDN denotes a dominant neighborhood defined in [20]. kDN can be viewed as a more robust version of k nearest neighborhood kNN. Different from kNN, kDN may contain less than k nearest neighbors. The parameter k is determined experimentally for each dataset presented below.

4.1 MPEG-7 Dataset

The proposed framework is tested for shape retrieval on a commonly used MPEG7 CE-Shape-1 part B database [33]. The dataset contains 1400 silhouette images from 70 classes, where each class has 20 different

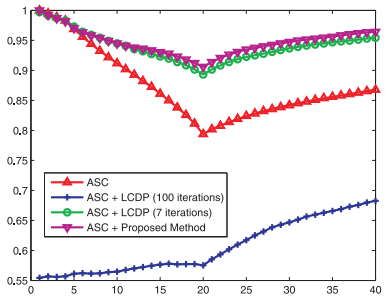

Fig. 6. The curves show the percentage of correct shapes (from the same class) among the first K most similar shapes for $K = 1, ..., 40$ on MPEG-7 shape dataset.

shapes. The retrieval rate is measured by the bull's eye score: every shape in the database is submitted as a query and the number of shapes from the same class in the top 40 is counted. The bull's eye score is then defined as the ratio of the number of correct hits to the best possible number of hits (which is 20×1400).

As shown in Table 2 the proposed affinity learning method can successfully improve on the state-of-the-art methods. We selected two different shape similarity methods: Aspect Shape Context (ASC) [9] and Articulated Invariant Representation (AIR) [34] as the input pairwise distance measure. kDN with $k = 10$ was used. We observe that the affinities learned by our method improve the original retrieval score of ASC by over 8%. We reach nearly perfect bull's eye score 99.99% on MPEG7 Dataset by using AIR for shape similarity. This is the best ever reported score on this popular shape dataset.

In order to visualize the gain in retrieval rates by our method, we plot the percentage of correct shapes (from the same class) among the first K most similar shapes for $K = 1, ..., 40$ in Fig. 6, where we use ASC for shape similarity. We observe that not only does the proposed method increase the bulls eye score, but also consistently achieves the best retrieval rates. Recall that each class has 20 shapes, which is the reason for the precision curves to increase for $k > 20$. In order to illustrate the problem with the stopping time of the graph classical diffusion computed by matrix power, we show two curves for LCDP [3], one when it is stopped after 7 iterations and the second one when it is stopped after 100 iterations, which clearly illustrates the problem of diffusing relevant information. In contrast, the proposed algorithm is robust to the number of iterations.

4.2 Nister and Stewenius (N-S) dataset

In this section, we demonstrate the performance of the proposed approach on image retrieval. We compare it to other diffusion based methods and to a recently proposed method, Contextual Dissimilarity Measure

Fig. 7. Some images from Nister and Stewenius (N-S) dataset.

Fig. 8. Sample images from our subset of Caltech 101 dataset.

(CDM) [18], which can significantly improve the similarity computed by bag-of-features. CDM learns affinities following a different principles than the proposed method. CDM is motivated by an observation that a good ranking is usually not symmetrical in image search. CDM makes two images similar when they both obtain a good ranking position when using each other as a query.

We selected the Nister and Stewenius (N-S) dataset [36] composed of 10,200 images. A few example images from N-S dataset are shown in Fig. 7. The N-S dataset consists of 2,550 objects or scenes, each of which is imaged from 4 different viewpoints. Hence there is only 4 images in each class and total of 2,550 image classes, which makes this dataset very challenging for any manifold learning approach, and in particular, for any diffusion based approach.

To obtain the pairwise distance relation between images for our algorithm, we implemented a baseline method described in [18]. The image descriptor is a combination of Hessian-Affine region detector [37] and SIFT descriptor [38]. A visual vocabulary is obtained using the k-means algorithm on the subsampled image descriptors.

The results are shown in Table 3. The retrieval rate is measured by the average number of correct images among the four first images returned. Thus, the maximum value is 4 and the higher the value the better is the result. Each image has been submitted as a query. The fact that our method can significantly improve the retrieval result of the baseline method (from 3.22 to 3.61) clearly shows the benefits of utilizing higher order relations by the TPG diffusion. We also observe that the result of our method is better than CDM. Finally, the usage of dominant neighborhood slightly improves on the result obtained with classic kNN. We did not have much choice to set the neighborhood size k for this dataset. kNN and kDN with $k = 3$ was used.

Since each image class has only 4 images, it is very difficult to correctly propagate the similarity relations. Therefore, the classic diffusion [8] can only improve the baseline result for a very small number of iterations. The best retrieval rate of the classic diffusion

TABLE 2
Retrieval rates (bull's eye) of different context shape retrieval methods on the MPEG-7 shape dataset.

IDSC [35]	IDSC +LP [5]	IDSC +Mutual graph [4]	Perc. R. [14]	Perc. R. + LCDP [14]	ASC [9]	ASC + LCDP [9]	ASC + TPG Diffusion	AIR [34]	AIR + TPG Diffusion
85.40%	91.61%	93.40%	88.39%	95.60%	88.30%	95.96%	**96.47%**	93.67%	**99.99%**

TABLE 3
Retrieval results on N-S Dataset. The highest possible score is 4.

Baseline [18]	Classic Diffusion with $t = 2$	Classic Diffusion with $t = 5$	Diffusion Maps [8]	CDM [18]	TPG Diffusion with Classic kNN	TPG Diffusion with DN
3.22	3.42	0.245	1.01	3.57	3.58	**3.61**

is for $t = 2$, i.e., when the original similarity matrix is raised to power $t = 2$. Already for $t = 5$, the retrieval rate is much lower than the rate of the baseline. We also report the retrieval result obtained after embedding the data by Diffusion Maps [8], which are significantly lower than the rate of the baseline. This justifies our observation that although Diffusion Maps are excellent for embedding into Euclidean spaces, the distances obtained after the embedding cannot be used for retrieval tasks.

4.3 Caltech 101 dataset

Besides N-S dataset, we also test our algorithm on a well known Caltech 101 dataset [39]. The Caltech-101 dataset contains 101 classes (including animals, vehicles, flowers, etc.) with high shape variability. The number of images per category varies from 31 to 800. Most images are medium resolution, i.e. about 300×200 pixels. We selected 12 classes from Caltech-101, which contain total 2788 images. Example images are shown in Fig.8. Different from experiments on N-S dataset, we just use pure SIFT descriptor [38] to calculate the distance between images. The SIFT features are extracted from 16×16 pixel patches densely sampled from each image on a grid with step size of 8 pixels. To get the codebook, we use standard K-means clustering and fix the codebook size to 2048. Each image is represented by multiple assignment [18] and Spatial Pyramid Matching [40]. The distance between two images is obtained by the χ^2 distance between the two vectors.

TABLE 4
Retrieval rates on 12 image classes from Caltech-101. The best possible rate is 1.

Baseline	Classic Dif. with $t = 5$	Classic Dif. with $t = 50$	Dif. Maps [8]	TPG Dif.
0.801	0.859	0.267	0.534	**0.903**

The results are shown in Table 4. It is clear that with adjusted number of iterations according to the ground truth, which is $t = 5$, the classic diffusion process is able to reveal the relation between images. However, as discussed above, it is very sensitive to number of iterations, which we illustrate with its retrieval rate for $t = 50$. Besides, the results of Diffusion Maps [8] demonstrates that the relation between objects after embedding by Diffusion Maps [8] is not suitable for retrieval. In our algorithm, kDN with $k = 400$ was used. Again TPG diffusion is able to significantly improve the retrieval rate of the input pairwise distance measure. In particular, this demonstrates that TPG diffusion is robust to large variance in the number of images in each class.

Fig. 9. Example images on INRIA Holiday Dataset.

4.4 INRIA Holidays Dataset

INRIA Holidays Dataset [41] contains 1491 high resolution images, 500 queries and 991 corresponding relevant images of mainly personal holiday photos. Some example images are shown in Fig. 9. The dataset is composed of 500 image groups, each of which represents a distinct scene. The first image of each group is the query image and the correct retrieval results are the other images of the group.

The authors of the dataset also provide a set of 4.45M precomputed SIFT descriptors and a visual dictionary with 200K visual words learned on Flickr60K. We used them to compute the baseline similarity between the images using the same function as in [42], [31], which corresponds to the bag-of words model with an additional inverse document frequency weighting term.

We use the standard mean average precision (mAP) to evaluate the performance, which is the mean of the average precision scores for each query. For each query, we are able to obtain a precision/recall curve. Then, the average precision score for this query is defined as the average value of precision value over the interval from recall equaling 0 to 1. It is equal to the area under the precision/recall curve.

The performance of the baseline similarity is $mAP = 50.3\%$. With setting $K = 20$, the learned similarities improve the results to $mAP = 68.5\%$. It is still not as good as some of the state of art retrieval results, [41], [31]. However, as we emphasize, we do not propose a stand alone image retrieval algorithm. We propose a generic affinity learning approach, which is able to learn better similarities.

5 Image Segmentation Performance

In this section we demonstrate that the learned affinities can significantly improve image segmentation. Here the data points are image regions, which are also called superpixels, and their initial affinities reflect their color similarity.

5.1 Hierarchical Graph Construction

The data points in this application represent superpixels obtained by different scales of a hierarchical image segmentation. We generate L hierarchical over-segmentation results by by varying the segmentation parameters of method [26], which are the input parameters to a Canny edge detector. V_l^h represents the regions at level h of the lth over-segmentation. In particular, V_l^1 denotes the regions of over-segmentation l at the finest level. Since we use the hierarchical segmentation, the constructed graph $G = (V, A)$ has the total of n nodes representing all segmentation regions in the set

$$V = V_1^1 \cup V_2^1 \cup \cdots \cup V_1^2 \cup V_2^2 \cup \cdots \cup V_L^H.$$

The edges are connected by different criteria according to the node types. Edge weight $a_{ij} \in A$ is non-zero if two regions i and j satisfy one of the following conditions

- regions i and j are at the finest level of some segmentation l, i.e., $i, j \in V_l^1$, and they share a common boundary, which we denote as $i \in Neighbor_l(j)$,
- regions i and j are at the finest level in two different segmentations l_1, l_2, i.e., $i \in V_{l_1}^1$ $j \in V_{l_2}^1$, and they overlap, which we denote as $Overlap(i, j)$
- in the same l segmentation and region j is the parent of region i, i.e., $i \in V_l^h$ and $j \in V_l^{h+1}$, which we will denote $j = Parent_l(i)$.

In these cases, edge weight $a_{ij} \in A$ is defined as

$$a_{ij} = \begin{cases} \exp\left(-\frac{||\bar{c_i} - \bar{c_j}||}{\delta}\right), & i \in Neighbor_l(j) \\ \exp\left(-\frac{||\bar{c_i} - \bar{c_j}||}{\delta}\right), & Overlap(i, j) \\ \lambda, & j = Parent_l(i) \end{cases} \quad (14)$$

where $\bar{c_i}$ represents the mean color of region i and λ controls the relation between regions in different layers. The larger λ is, the more information can be propagated following the hierarchical structure of regions. The smaller δ is, the more sensitive is the edge weight to the color differences. In all our experiments, we set $\lambda = 0.0001$ and $\delta = 60$.

5.2 Final Segmentation

The graph $G = (V, A)$ constructed in Section 5.1 is the input to the iterative algorithm in Section 3.2. The learned affinities in in matrix A^* are the input to a final image segmentation presented in this section. We consider the segmentation as a labeling problem in which one label $k \in \{1, \cdots, K\}$ is assigned to each node i.

Let $\vec{y_k} = [y_{ik}]$ be a $n \times 1$ partitioning (or indicator) vector with $y_{ik} = 1$ if i belongs to the k-th cluster and 0 otherwise. The clustering criterion follows the standard Normalized Cuts [12] procedure:

$$\text{maximize } S(Y) = \frac{1}{K} \sum_{k=1}^{K} \frac{\vec{y_k}^T A^* \vec{y_k}}{\vec{y_k}^T D \vec{y_k}}. \quad (15)$$

subject to $YY^T = I$, where $Y = [\vec{y}_1, \cdots, \vec{y}_K]$. $D = diag([d_1, \cdots, d_n])$ is the degree matrix, where $d_i = \sum_{j=1}^{n} A^*(i, j)$. The optimal solution of $S(Y)$ is the subspace spanned by the K largest eigenvectors of the matrix B [21]:

$$B = D^{-\frac{1}{2}} A^* D^{-\frac{1}{2}} \quad (16)$$

B is a $n \times n$ matrix, which contains information for all the regions. Following the standard procedure we obtain a K dimensional vector in the subspace spanned by the K largest eigenvectors for each image region given as node in V.

Since to obtain the final image segmentation, we only need one set of superpixels, we select the set of superpixels $V_{l_m}^1$ with the largest number of regions on the finer segmentation level. Finally, we run standard K-means clustering on the K dimensional vectors representing the regions in $V_{l_m}^1$. The obtained K cluster labels are assigned to the regions in $V_{l_m}^1$ to obtain the final segmentation result.

Below we report on extensive evaluation to illustrate the benefits of learned similarities for unsupervised image segmentation. We show both qualitative and quantitative segmentation results on two publicly available natural image databases: Berkeley Segmentation Dataset (BSDS)[43] and the MSRC Object Recognition Dataset (MSRC) [44].

For qualitative evaluation, two different measures are used. The earliest and most obvious measure is Probabilistic Rand Index (PRI) [45], which counts the number of pixel pairs whose labels are consistent between the segmentation and the ground truth. PRI is also the most natural measure, since it measures the accuracy of the segmentation label assignment. Variation of Information (VOI) [46] defines the distance between two segmentations as the average conditional entropy of one segmentation given the other, and thus roughly measures the amount of randomness in one segmentation which cannot be explained by the other. We choose these two measurements, since all the other papers have been evaluated according to them. A

Methods	PRI	VOI
NC [12]	0.7242	2.9061
MS [13]	0.7958	1.9725
F-H [47]	0.7139	3.3949
NTP [48]	0.7521	2.4954
Saliency [49]	0.7758	1.8165
LAS [21]	0.8146	1.8545
gpb-owt-ucm [25]	0.81	**1.68**
TBES [50]	0.80	1.76
TPGD without Hierarchy	0.8032	1.925
TPGD + Hierarchy	**0.8227**	1.7696

TABLE 5
Quantitative comparison of our algorithm to other segmentation methods over the BSDS. A segmentation result is viewed as better if PRI is larger and VOI is smaller.

segmentation result is viewed as better if PRI is larger and VOI is smaller.

We use two segmentation results with different parameters for each dataset. We use three levels of segmentation results, from finest to coarse. To get the optimal performance of our algorithm, we follow others strategy [21] to determine the number of segments by comparing to the labeled image.

5.3 Berkeley Segmentation Dataset

The Berkeley Segmentation Dataset consists of 300 natural images, each of which has been hand segmented by multiple human subjects. We compare the performance of our method to several publicly available image segmentation methods, which we refer to as NC [12], MS [13], gpb-owt-UCM [25], F-H [47], NTP [48], Saliency [49], TBES [50], and LAS [21]. The reported results of these methods are cited from the above papers, respectively. We also report the results of TPG without using the hierarchical structure. Table 5 summarizes the performance of our method and compares it to other methods. Our segmentation obtains best score on PRI, which means that our segmentation label assignment is most accurate. Its scores according to VOI is among top 3 results. Fig. 10 illustrates some representative segmentation results.

5.4 MSRC Object Recognition Dataset

The MSRC Object Recognition Database consists of 591 images of objects grouped into 20 categories. We evaluate the performance using the cleaned up ground-truth labeling provided by the authors of [51]. This dataset is highly challenging because, in many cases, the ground truth segmentation only draws a boundary around the salient object in the image, casting everything else as background. The results and comparison with TBES segmentation are shown in Table 6. The other algorithms did not report their results on this dataset. The proposed algorithm reaches excellent performance, and outperforms TBES on PRI

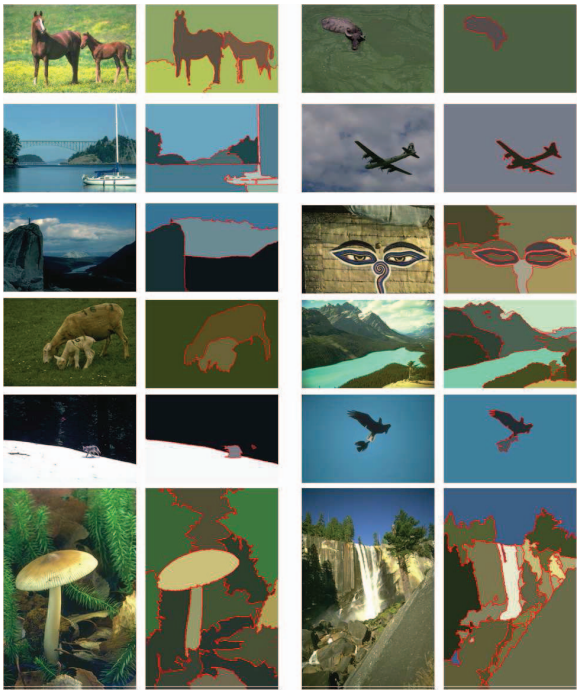

Fig. 10. Representative Examples on BSDS Dataset. The image in the odd columns are original images and the images in the even columns are the segmentation results. The color in each region is the average color over the pixels in the region.

Methods	PRI	VOI
TBES [50]	0.76	1.49
TPGD + Hierarchy	**0.8140**	**1.2564**

TABLE 6
Quantitative comparison of our algorithm with TBES segmentation on MSRC dataset.

and VOI measures. Some representative segmentation results of our algorithm are shown in Fig. 12.

We also test our algorithm on the original dataset, which contains void regions in the ground-truth labeling [44]. Some examples of cleaned up ground truth images and the noisy ones with void regions are shown in Figure 11. The void regions (white regions) are often have irregular shape and are randomly located in the images. The PRI and VOI scores of the proposed algorithm on the noisy ground truth are 0.7724 and 1.5758 respectively. Since the ground truth is noisy, the results are lower than the ones we obtain based on the cleaned ground truth image. However, the PRI score is still better than TBES (Table 6) on the cleaned ground truth images.

6 Conclusion

The key advantage of the proposed Tensor Product Graph diffusion is the utilization of higher order similarity relations, which are both local and long

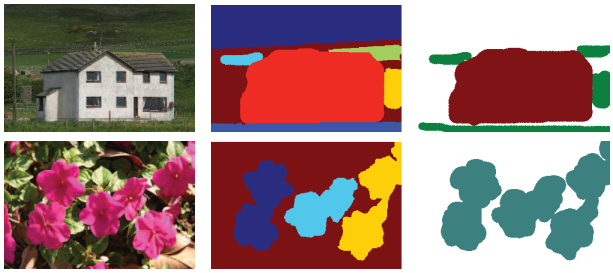

Fig. 11. Representative examples on cleaned up and noisy groundtruth images of MSRC dataset. First column: original images. Second column: cleaned up ground truth images. Third column: noisy ground truth images with void regions shown in white.

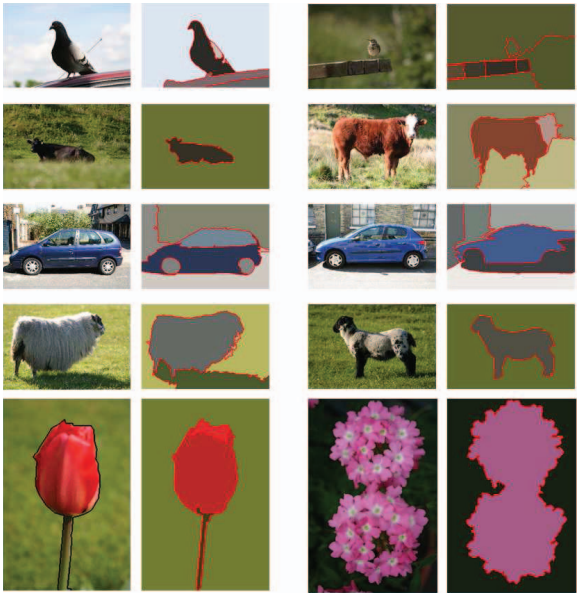

Fig. 12. Representative Examples on MSRC Dataset. The image in the odd columns are original images and the images in the even columns are the segmentation results. The color in each region is the average color over the pixels in the region.

range. Usually higher order relations lead to a substantially higher computation cost. However, we are able to introduce an iterative algorithm to compute TPG diffusion that has the same space and time complexity as the classical diffusion on the original graph. We also provide a formal proof that the iterative algorithm and the TPG diffusion converge to the same solution. Hence the proposed TPG diffusion explores the benefits of higher order relations without the price of higher computational cost. As shown in our experimental results, the new affinities learned with the TPG diffusion lead to good performance in image retrieval and in image segmentation.

ACKNOWLEDGMENTS

The work was supported by the DOE Award 71498-001-09 and by the NSF under Grants IIS-0812118, BCS-0924164, OIA-1027897.

REFERENCES

[1] B. Schölkopf, A. Smola, and K.-R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Computation*, vol. 10(5), pp. 1299–1319, 1998.
[2] D. Zhou, J. Weston, A.Gretton, Q.Bousquet, and B.Scholkopf., "Ranking on data manifolds," in *NIPS*, 2003.
[3] X. Yang, S. Köknar-Tezel, and L. J. Latecki, "Locally constrained diffusion process on locally densified distance spaces with applications to shape retrieval," in *CVPR*, 2009.
[4] P. Kontschieder, M. Donoser, and H. Bischof, "Beyond pairwise shape similarity analysis," in *ACCV*, 2009.
[5] X. Bai, X. Yang, L. J. Latecki, W. Liu, and Z. Tu, "Learning context sensitive shape similarity by graph transduction," *IEEE Trans. on PAMI*, 2010.
[6] M. Szummer and T. Jaakkola, "Partially labeled classification with markov random walks," in *NIPS*, 2001.
[7] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Scholkopf., "Learning with local and global consistency," in *NIPS*, 2003.
[8] R. Coifman and S. Lafon, "Diffusion maps," *Applied and Computational Harmonic Analysis*, vol. 21, pp. 5–30, 2006.
[9] H. Ling, X. Yang, and L. J. Latecki, "Balancing deformability and discriminability for shape matching," in *ECCV*, 2010.
[10] Y. Huang, Q. Liu, and D. Metaxas, "Video object segmentation by hypergraph cut," in *CVPR*, 2009.
[11] D. Zhou, J. Huang, and B. Schlkopf, "Learning with hypergraphs: Clustering, classification, and embedding," in *NIPS*, 2007.
[12] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE PAMI*, vol. 22, pp. 888–905, 2000.
[13] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE PAMI*, 2002.
[14] A. Temlyakov, B. C. Munsell, J. W. Waggoner1, and S. Wang, "Two perceptually motivated strategies for shape classification," in *CVPR*, 2010.
[15] X. Bai, B. Wang, X. Wang, W. Liu, and Z. Tu, "Co-transduction for shape retrieval," in *ECCV*, 2010.
[16] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," in *CVPR*, 2006.
[17] J. Sivi and A. Zisserman, "Video google: A text retrieval approach toobject matching in videos," in *ICCV*, 2003.
[18] H. Jegou, C. Schmid, H. Harzallah, and J. Verbeek, "Accurate image search using the contextual dissimilarity measure," *IEEE PAMI*, 2010.
[19] H. Jegou, M. Douze, and C. Schmid, "Improving bag-of-features for large scale image search," *IJCV*, vol. 87, pp. 191–212, 2010.
[20] X. Yang and L. J. Latecki, "Affinity learning on a tensor product graph with applications to shape and image retrieval," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2011, pp. 2369–2376.
[21] T. H. Kim, K. M. Lee, and S. U. Lee, "Learning full pairwise affinities for spectral segmentation," in *CVPR*, 2010.
[22] L. Ladicky, C. Russell, and P. Kohli, "Associative hierarchical crfs for object class image segmentation," in *ICCV*, 2009.
[23] N. Plath, M. Toussaint, and S. Nakajima, "Multi-class image segmentation using conditional random fields and global classification," in *ICML*, 2009.
[24] J. J. Lim, P. Arbelaez, C. Gu, and J. Malik, "Context by region ancestry," in *ICCV*, 2009.
[25] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "From contours to regions: An empirical evaluation," in *CVPR*, 2009.
[26] L. Prasad and S. Swaminarayan, "Hierarchical image segmentation by polygon grouping," in *CVPR 2008 Workshop on Perceptual Organization in Computer Vision*, 2008.
[27] X. Ren and J. Malik, "Learning a classification model for segmentation," in *ICCV*, 2003.

[28] S. Lafon and A. B. Lee, "Diffusion maps and coarse-graining: A unified framework for dimensionality reduction graph partitioning, and data set parameterization," *IEEE PAMI*, vol. 28, pp. 1393–1403, 2006.
[29] P. Lancaster and L. Rodman, *Algebraic Riccati Equations*. Oxford: Clarendon Press, 1995.
[30] C. van Loan, "The ubiquitous kronecker product," *J. of Computational and Applied Math.*, vol. 123, pp. 85–100, 2000.
[31] D. Qin, S. Gammeter, L. Bossard, and T. Q. L. van Gool, "Hello neighbor: accurate object retrieval with k-reciprocal nearest neighbors," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2011.
[32] J. Wang, S.-F. Chang, X. Zhou, and T. C. S. Wong, "Active microscopic cellular image annotation by superposable graph transduction with imbalanced labels," in *CVPR*, 2008.
[33] L. J. Latecki and R. Lakämper, "Shape similarity measure based on correspondence of visual parts," *IEEE Trans. PAMI*, vol. 22(10), pp. 1185–1190, 2000.
[34] R. Gopalan, P. Turaga, and R. Chellappa, "Articulation-invariant representation of non-planar shapes," in *ECCV*, 2010.
[35] H. Ling and D. Jacobs, "Shape classification using the inner-distance," *IEEE Trans. PAMI*, vol. 29, pp. 286–299, 2007.
[36] H. Stewenius and D. Nister, "Object recognition benchmark. http://vis.uky.edu/ stewe/ukbench/."
[37] K. Mikolajczyk and C. Schmid, "Scale and affine invariant interest point detectors," *IJCV*, vol. 60(1), pp. 63–86, 2004.
[38] D. Lowe, "Distinctive image features from scale-invariant key points." *IJCV*, vol. 60, pp. 91–110, 2004.
[39] L. Fei-Fei, R. Fergus, and P. Perona, "One-shot learning of object categories," *IEEE PAMI*, vol. 28, pp. 594–611, 206.
[40] S. Lazebnik, C. Schimid, and J. Ponce, "Beyong bags of features: Spatial pyramid matching for recognizing natural scene categories," in *CVPR*, 2006.
[41] M. D. Herve Jegou and C. Schmid, "Hamming embedding and weak geometry consistency for large scale image search," in *ECCV*, 2008.
[42] H. Jegou, M. Douze, and C. Schmid, "Hamming embedding and weak geometric consistency for large scale image search," in *ECCV*, 2008.
[43] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *ICCV*, 2001.
[44] J. Shotton, J. Winn, C. Rother, and A. Criminisi, "Texton-boost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation." in *ECCV*, 2006.
[45] W. M. Rand, "Objective criteria for the evaluation of clustering methods," *Journal of the American Statistical Association*, vol. 66, pp. 846–850, 1971.
[46] M. Meila, "Comparing clusterings: an axiomatic view." in *ICML*, 2005.
[47] P. Felzenszwalb and D. Huttenlocher, "Efficient graph-based segmentation algorithm," *IJCV*, 2004.
[48] J. Wang, Y. Jia, X.-S. Hua, C. Zhang, and L. Quan, "Normalized tree partitioning for image segmentation," in *CVPR*, 2008.
[49] M. Donoser, M. Urschler, M. Hirzer, and H. Bischof, "Saliency driven total variation segmentation," in *ICCV*, 2009.
[50] S. R. Rao, H. Mobahi, A. Y. Yang, S. S. Sastry, and Y. Ma, "Natural image segmentation with adaptive texture and boundary encoding," in *ACCV*, 2009.
[51] T. Malisiewicz and A. A. Efros, "Improving spatial support for objects via multiple segmentations." in *BMVC*, 2007.

Xingwei Yang received the B.E. degree in electronics and information engineering from Huazhong University of Science and Technology (HUST), Wuhan, China, in 2002. Currently, he works as Research Scientist in GE Global Research Center after he obtained his Ph.D. at Temple University in 2011.

Lakshman Prasad is with the Data Exploitation team in the Intelligence and Space Research (ISR) division of Los Alamos National Laboratory. He received his MS in Mathematics from the Indian Institute of Technology, Kanpur, India, and a PhD In Computer Science from Louisiana State University, Baton Rouge, Louisiana. His interests include Image Analysis, Computer Vision, Computational Geometry, and Multiscale Data Analysis.

Longin Jan Latecki received the PhD degree in computer science from Hamburg University, Germany, in 1992. He is a professor of computer science at Temple University, Philadelphia. His main research interests include shape representation and similarity, object detection and recognition in images, robot perception, machine learning, and digital geometry. He has published 200 research papers and books. He is an editorial board member of Pattern Recognition and International Journal of Mathematical Imaging. He received the annual Pattern Recognition Society Award together with Azriel Rosenfeld for the best article published in the journal Pattern Recognition in 1998. He is the recipient of the 2000 Olympus Prize, the main annual award, from the German Society for Pattern Recognition (DAGM).