Learning Context Sensitive Shape Similarity by Graph Transduction

Xiang Bai[1,3] Xingwei Yang[2], Longin Jan Latecki[2] Wenyu Liu[1], and Zhuowen Tu[3]

[1] Dept. of Electronics and Information Engineering, Huazhong University of Science and Technology, Wuhan, China

xiang.bai@gmail.com, liuwy@hust.edu.cn

[2] Dept. of Computer and Information Sciences, Temple University, Philadelphia

{xingwei.yang, latecki}@temple.edu

[3] Lab of Neuro Imaging, University of California, Los Angeles

zhuowen.tu@loni.ucla.edu

Abstract

Shape similarity and shape retrieval are very important topics in computer vision. The recent progress in this domain has been mostly driven by designing smart shape descriptors for providing better similarity measure between pairs of shapes. In this paper, we provide a new perspective to this problem by considering the existing shapes as a group, and study their similarity measures to the query shape in a graph structure. Our method is general and can be built on top of any existing shape similarity measures. For a given similarity measure s_0, a new similarity s is learned through graph transduction. Intuitively, for a given query shape q, the similarity $s(q, p)$ will be high if neighbors of p are also similar to q. However, even if $s_0(q, p)$ is very high, but the neighbors of p are not similar to q, then $s(q, p)$ will be low. The new similarity is learned iteratively so that the neighbors of a given shape influence its final similarity to the query. The basic idea here is related to PageRank ranking, which forms a foundation of google web search. The presented experimental results demonstrate that the proposed approach yields significant improvements over the state-of-art shape matching algorithms. We obtained a retrieval rate of **91%** on the MPEG-7 data set, which is the highest ever reported in the literature. Moreover, the learned similarity by the proposed method also achieves the promising improvements on both shape classification and shape clustering.

Index Terms

Shape matching, shape retrieval, shape classification, shape clustering, graph transduction

I. INTRODUCTION

Shape matching/retrieval is a very critical problem in computer vision. There are many different kinds of shape matching methods, and the progress in increasing the matching rate has been substantial in recent years. However, all of these approaches are focused on the nature of shape similarity. It seems to be an obvious statement that the more similar two shapes are, the smaller is their difference, which is measured by some distance function. Yet, this statement ignores the fact that some differences are relevant while other differences are irrelevant for shape similarity. It is not yet clear how the biological vision systems perform shape matching; it is clear that shape matching involves the high-level understanding of shapes. In particular, shapes in the same class can differ significantly because of distortion or non-rigid transformation. In other words, even if two shapes belong to the same class, the distance between them may be very large if the distance measure cannot capture the intrinsic property of the shape. It appears to us that

all published shape distance measures [1]–[11] are unable to address this issue. For example, based on the inner distance shape context (IDSC) [3], the shape in Fig. 1(a) is more similar to (b) than to (c), but it is obvious that shape (a) and (c) belong to the same class. This incorrect result is due to the fact that the inner distance is unaware that the missing tail and one front leg are irrelevant for this shape similarity judgment. On the other hand, much smaller shape details like the dog's ear and the shape of the head are of high relevance here. No matter how good a shape matching algorithm is, the problem of relevant and irrelevant shape differences must be addressed if we want to obtain human-like performance. This requires having a model to capture the essence of a shape class instead of viewing each shape as a set of points or a parameterized function.

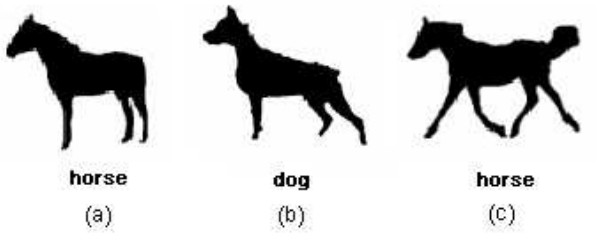


Fig. 1. Existing shape similarity methods incorrectly rank shape (b) as more similar to (a) than (c).

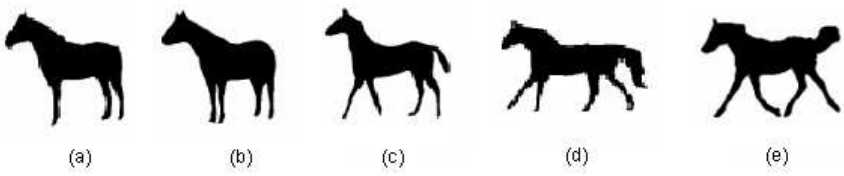


Fig. 2. A key idea of the proposed distance learning is to replace the original shape distance between (a) and (e) with a geodesic path in the manifold of know shapes, which is the path (a)-(e) in this figure.

In this paper, we propose to use a graph-based transductive learning algorithm to tackle this problem, and it has the following properties: (1) Instead of focusing on computing the distance (similarity) for a pair of shapes, we take advantage of the manifold formed by the existing shapes. (2) However, we do not explicitly learn the manifold nor compute the geodesics [12], which are

time consuming to calculate. A better metric is learned by collectively propagating the similarity measures to the query shape and between the existing shapes through graph transduction. (3) Unlike the label propagation [13] approach, which is semi-supervised, we treat shape retrieval as an unsupervised problem and do not require knowing any shape labels. (4) We can build our algorithm on top of any existing shape matching algorithm and a significant gain in retrieval rates can be observed on well-known shape datasets.

Given a database of shapes, a query shape, and a shape distance function, which does not need to be a metric, we learn a new distance function that is expressed by shortest paths on the manifold formed by the know shapes and the query shape. We can do this without explicitly learning this manifold. As we will demonstrate in our experimental results, the new learned distance function is able to incorporate the knowledge of relevant and irrelevant shape differences. It is learned in an unsupervised setting in the context of known shapes. For example, if the database of known shapes contains shapes (a)-(e) in Fig. 2, then the new learned distance function will rank correctly the shape in Fig. 1(a) as more similar to (c) than to (b). The reason is that the new distance function will replace the original distance (a) to (c) in Fig.1 with a distance induced by the shortest path between in (a) and (e) in Fig.2.

In more general terms, even if the difference between shape A and shape C is large, but there is a shape B which has small difference to both of them, we still claim that shape A and shape C are similar to each other. This situation is possible for most shape distances, since they do not obey the triangle inequality, i.e., it is not true that $d(A,C) \leq d(A,B) + d(B,C)$ for all shapes A, B, C [14]. We propose a learning method to modify the original shape distance $d(A,C)$. If we have the situation that $d(A,C) > d(A,B) + d(B,C)$ for some shapes A, B, C, then the proposed method is able to learn a new distance $d'(A,C)$ such that $d'(A,C) \leq d(A,B) + d(B,C)$. Further, if there is a path in the distance space such that $d(A,C) > d(A,B_1) + \ldots + d(B_k,C)$, then our method learns a new $d'(A,C)$ such that $d'(A,C) \leq d(A,B_1) + \ldots + d(B_k,C)$. Since this path represents a minimal distortion morphing of shape A to shape C, we are able to ignore irrelevant shape differences, and consequently, we can focus on relevant shape differences with the new distance d'.

Our experimental results clearly demonstrate that the proposed method can improve the retrieval results of the existing shape matching methods. We obtained the retrieval rate of **91%** on part B of the MPEG-7 Core Experiment CE-Shape-1 data set [15], which is the highest ever

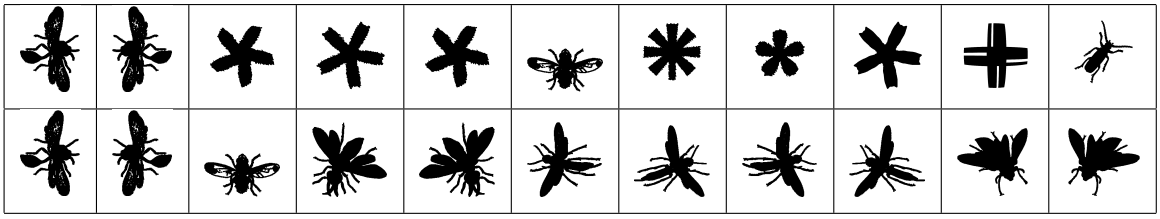


Fig. 3. The first column shows the query shape. The remaining 10 columns show the most similar shapes retrieved from the MPEG-7 data set. The first row shows the results of IDSC [3]. The second row shows the results of the proposed learned distance.

bull's eye score reported in the literature. As the input to our method we used the IDSC, which has the retrieval rate of 85.40% on the MPEG-7 data set [3]. Fig. 3 illustrates the benefits of the proposed distance learning method. The first row shows the query shape followed by the first 10 shapes retrieved using IDSC only. Only two flies are retrieved among the first 10 shapes. The results of the learned distance for the same query are shown in the second row. All of the top 10 retrieval results are correct. The proposed method was able to learn that the shape differences in the number of fly legs and their shapes are irrelevant. A preliminary version of this paper appeared as [16]. The remainder of this paper is organized as follows. In Section II, we briefly review some well-known shape matching methods and the semi-supervised learning algorithms. Section III describes the proposed approach to learning shape distances. Section IV relates the proposed approach to the class of machine learning approaches called label propagation. The problem of the construction of the affinity matrix is addressed in Section V. Besides, a novel shape clustering algorithm is introduced in Section VI. Section VII gives the experimental results on several famous shape data sets to show the advantage of the proposed approach. Conclusion and discussion are given in Section VIII.

II. Related work

The semi-supervised learning problem has attracted an increasing amount of interest recently, and several novel approaches have been proposed. The existing approaches could be divided into several types, multiview learning [17], generative model [18], Transductive Support Vector Machine (TSVM) [19]. Recently there have been some promising graph based transductive learning approaches proposed, such as label propagation [13], Gaussian fields and harmonic functions (GFHF) [20], local and global consistency (LGC) [21], and the Linear Neighborhood

Propagation (LNP) [22]. Zhou et al. [23] modified the LGC for the information retrieval. The semi-supervised learning problem is related to manifold learning approaches, e.g., [24].

The proposed method is inspired by the label propagation. The reason we choose the framework of label propagation is it allows the clamping of labels. Since the query shape is the only labeled shape in the retrieval process, the label propagation allows us to enforce its label during each iteration, which naturally fits in the framework of shape retrieval. Usually, GFHF is used instead of label propagation, as both methods can achieve the same results [13]. However, in the shape retrieval, we can use only the label propagation, the reason is explained in detail in Section IV.

Since a large number of shape similarity methods have been proposed in the literature, we focus our attention on methods that reported retrieval results on the MPEG-7 shape data set (part B of the MPEG-7 Core Experiment CE-Shape-1) [15]. This allows us to clearly demonstrate the retrieval rate improvements obtained by the proposed method. Belongie et al. [1] introduced a novel 2D histograms representation of shapes called Shape Contexts (SC). Ling and Jacobs [3] modified the Shape Context by considering the geodesic distance between contour points instead of the Euclidean distance, which significantly improved the retrieval and classification of articulated shapes. Latecki and Lakäemper [4] used visual parts represented by simplified polygons of contours for shape matching. Tu and Yuille [2] combined regions and contours together within a generative model for shape matching. In order to avoid problems associated with purely global or local methods, Felzenszwalb and Schwartz [7] described a dynamic and hierarchical curve matching method. Other hierarchical methods include the hierarchical graphical models in [25] and hierarchical procrustes matching [6]. Alajlan et al. proposed a mutiscale representation of triangle areas for shape matching, which also included partial and global shape information [26]. Daliri and Torre defined a symbolic descriptor based on Shape Contexts, then used edit distance for final matching in order to overcome the difficulty caused by deformation and occlusions [27]. The methods above all focused on designing nice shape descriptors or representation. Although the researchers started to integrate the global and partial shape similarity recently and achieved some progress, the improvement was not obvious as shown in Table I of Section VII (In this table, we summarized all the reported retrieval results on MPEG-7 database, and the retrieval rates of the recent publications are all around 85%). There are two main reasons that limit the progress in shape retrieval: 1) The case for large deformation

and occlusions still can not be solved well. 2) The existing algorithms can not distinguish the relevant and irrelevant shape differences, which has been pointed out by us in Section I.

There is a significant body of work on distance learning [28]. Xing et al. [29] propose estimating the matrix W of a Mahalanobis distance by solving a convex optimization problem. Bar-Hillel et al. [30] also use a weight matrix W to estimate the distance by relevant component analysis (RCA). Athitsos et al. [31] proposed a method called BoostMap to estimate a distance that approximates a certain distance. Hertz's work [32] uses AdaBoost to estimate a distance function in a product space, whereas the weak classifier minimizes an error in the original feature space. All these methods' focus is a selection of suitable distance from a given set of distance measures. Our method aims at learning new distance to improve the retrieval and clustering performance.

III. Learning New Distance Measures

We first describe the classical setting of similarity retrieval. It applies to many retrieval scenarios like key word, document, image, and shape retrieval. Given is a set of objects $X = \{x_1, \ldots, x_n\}$ and a similarity function sim: $X \times X \rightarrow R^+$ that assigns a similarity value (a positive integer) to each pair of objects.

We assume that x_1 is a query object (e.g., a query shape), $\{x_2, \ldots, x_n\}$ is a set of known database objects (or a training set). Then by sorting the values $sim(x_1, x_i)$ in decreasing order for $i = 2, \ldots, n$ we obtain a ranking of database objects according to their similarity to the query, i.e., the most similar database object has the highest value and is listed first. Sometimes a distance measure is used in place of the similarity measure, in which case the ranking is obtained by sorting the database objects in the increasing order, i.e., the object with the smallest value is listed first. Usually, the first $N \ll n$ objects are returned as the most similar to the query x_1.

As discussed above, the problem is that the similarity function sim is not perfect so that for many pairs of objects it returns wrong results, although it may return correct scores for most pairs. We introduce now a method to learn a new similarity function sim_T that drastically improves the retrieval results of sim for the given query x_1.

Let $w_{i,j} = sim(x_i, x_j)$, for $i, j = 1, \ldots, n$, be a similarity matrix, which is also called an affinity matrix. We also define a $n \times n$ probabilistic transition matrix P as a row-wise normalized

matrix w.

$$P_{ij} = \frac{w_{ij}}{\sum_{k=1}^{n} w_{ik}} \tag{1}$$

where P_{ij} is the probability of transit from node i to node j.

We seek a new similarity measure s. Since s only needs to be defined as similarity of other elements to query x_1, we denote $f(x_i) = s(x_1, x_i)$ for $i = 1, \ldots, n$. We formulate now a key equation of the proposed approach. We seek a function f that satisfies

$$f(x_i) = \sum_{j=1}^{n} P_{ij}\ f(x_j) \tag{2}$$

Thus, the similarity of x_i to the query x_1, expressed as $f(x_i)$, is a weighted average over all other database objects, where the weights sum to one and are proportional to the similarity of the other database objects to x_i. In other words we seek a function $f : X \to [0, 1]$ such that $f(x_i)$ is a weighted average of $f(x_j)$, where the weights are based on the original similarities $w_{i,j} = sim(x_i, x_j)$. Our intuition is that the new similarity $f(x_i) = s(x_1, x_i)$ will be large iff all points x_j that are very similar to x_i (large $sim(x_i, x_j)$) are also very similar to query x_1 (large $sim(x_1, x_j)$).

The recursive equation (2) is closely related to PageRank. As stated in [33], a slightly simplified version of simple ranking R of a web page u in PageRank is defined as

$$R(u) = \sum_{v \in B_u} \frac{c}{N_v} R(v), \tag{3}$$

where B_u is a set of pages that point to u, N_v is the number of links from page v and c is a normalization factor.

Consequently, our equation (2) differers from PageRank equation (3) by the normalization matrix, which is defined in Eq. (1) in our case, and is equal to $\frac{c}{N_v}$ for PageRank. The PageRank recursive equation takes a simple average over neighbors (a set of pages that point to a given web page), while we take a weighted average over the original input similarities. Therefore, our equation admits recursive solution analog to the solution of the PageRank equation. Before we present it, we point out one more relation to recently proposed label propagation [13].

Label propagation belongs to a set of semi-supervised learning methods, where it is usually assumed that class labels are known for a small set of data points. We have an extreme case of semi-supervised learning, since we only assume that the class label of the query is known. Thus,

we have only one class that contains only one labeled element being the query x_1. We define a sequence of labeling functions $f_t : X \to [0,1]$ with $f_0(x_1) = 1$ and $f_0(x_i) = 0$ for $i = 2, \ldots, n$, where $f_t(x_i)$ can be interpreted as probability that point x_i has the class label of the query x_1.

We obtain the solution to Eq. (2) by the following recursive procedure:

$$f_{t+1}(x_i) = \sum_{j=1}^{n} P_{ij} \; f_t(x_j) \tag{4}$$

for $i = 2, \ldots, n$ and we set

$$f_{t+1}(x_1) = 1. \tag{5}$$

We define a sequence of new learned similarity functions restricted to x_1 as

$$sim_t(x_1, x_i) = f_t(x_i). \tag{6}$$

Thus, we interpret f_t as a set of normalized similarity values to the query x_1. Observe that $sim_1(x_1, x_i) = w_{1,i} = sim(x_1, x_i)$.

We iterate steps (4) and (5) until the step $t = T$ for which the change is below a small threshold. We then rank the similarity to the query x_1 with $s = sim_T$. Our experimental results in Section VII demonstrate that the replacement of the original similarity measure sim with sim_T results in a significant increase in the retrieval rate.

The steps (4) and (5) are used in label propagation, which is described in Section IV. However, our goal and our setting are different. Although label propagation is an instance of semi-supervised learning, we stress that we remain in the unsupervised learning setting. In particular, we deal with the case of only one known class, which is the class of the query object. This means, in particular, that label propagation has a trivial solution in our case $\lim_{t\to\infty} f_t(x_i) = 1$ for all $i = 1, \ldots, n$, i.e., all objects will be assigned the class label of the query shape. Since our goal is ranking of the database objects according to their similarity to the query, we stop the computation after a suitable number of iterations $t = T$. As is the usual practice with iterative processes that are guaranteed to converge, the computation is halted if the difference $||f_{t+1} - f_t||$ becomes very slow, see Section VII for details.

If the database of known objects is large, the computation with all n objects may become impractical. Therefore, in practice, we construct the matrix w using only the first $M < n$ most similar objects to the query x_1 sorted according to the original distance function sim.

IV. Relation to Label Propagation

Label propagation is formulated as a form of propagation on a graph, where node's label propagates to neighboring nodes according to their proximity. In our approach we only have one labeled node, which is the query shape. The key idea is that its label propagates "faster" along a geodesic path on the manifold spanned by the set of known shapes than by direct connections. While following a geodesic path, the obtained new similarity measure learns to ignore irrelevant shape differences. Therefore, when learning is complete, it is able to focus on relevant shape differences. We review now the key steps of label propagation and relate them to the proposed method introduced in Section III.

Let $\{(x_1, y_1) \dots (x_l, y_l)\}$ be the labeled data, $y \in \{1 \dots C\}$, and $\{x_{l+1} \dots x_{l+u}\}$ the unlabeled data, usually $l \ll u$. Let $n = l + u$. We will often use L and U to denote labeled and unlabeled data respectively. The Label propagation supposes the number of classes C is known, and all classes are present in the labeled data [13]. A graph is created where the nodes are all the data points, the edge between nodes i, j represents their similarity $w_{i,j}$. Larger edge weights allow labels to travel through more easily. Also define a $l \times C$ label matrix Y_L, whose ith row is an indicator vector for y_i, $i \in L$: $Y_{ic} = \delta(y_{i,c})$. The label propagation computes soft labels f for nodes, where f is a $n \times C$ matrix whose rows can be interpreted as the probability distributions over labels. The initialization of f is not important. The label propagation algorithm is as follows:

1) Initially, set $f(x_i) = y_i$ for $i = 1, \dots, l$ and $f(x_j)$ arbitrarily (e.g., 0) for $x_j \in X_u$
2) Repeat until convergence: Set $f(x_i) = \sum_{j=1}^{n} P_{ij}\ f(x_j)$, $\forall x_i \in X_u$ and
3) set $f(x_i) = y_i$ for $i = 1, \dots, l$ (the labels of the labeled objects should be fixed).

In step 2, all nodes propagate their labels to their neighbors for one step. Step 3 is critical, since it ensures persistent label sources from labeled data. Hence instead of letting the initial labels fade way, we fix the labeled data. This constant push from labeled nodes, helps to push the class boundaries through high density regions so that they can settle in low density gaps. If this structure of data fits the classification goal, then the algorithm can use unlabeled data to improve learning.

Let $f = \begin{pmatrix} f_L \\ f_U \end{pmatrix}$. Since f_L is fixed to Y_L, we are solely interested in f_U. The matrix P is split

into labeled and unlabeled sub-matrices

$$P = \begin{bmatrix} P_{LL} & P_{LU} \\ P_{UL} & P_{UU} \end{bmatrix} \tag{7}$$

As proven in [13] the label propagation converges, and the solution can be computed in closed form using matrix algebra:

$$f_U = (I - P_{UU})^{-1} P_{UL} Y_L \tag{8}$$

However, as the label propagation requires all classes be present in the labeled data, it is not suitable for shape retrieval. As mentioned in Section III, for shape retrieval, the query shape is considered as the only labeled data and all other shapes are the unlabeled data. Moreover, the graph among all of the shapes is fully connected, which means the label could be propagated on the whole graph. If we iterate the label propagation infinite times, all of the data will have the same label, which is not our goal. Therefore, we stop the computation after a suitable number of iterations $t = T$.

V. The Affinity Matrix

In this section, we address the problem of the construction of the affinity matrix W. There are some methods that address this issue, such as local scaling [34], local liner approximation [22], and adaptive kernel size selection [35].

However, in the case of shape similarity retrieval, a distance function is usually defined, e.g., [1], [3], [4], [7]. Let $D = (D_{ij})$ be a distance matrix computed by some shape distance function. Our goal is to convert it to a similarity measure in order to construct an affinity matrix W. Usually, this can be done by using a Gaussian kernel:

$$w_{ij} = \exp(-\frac{D_{ij}^2}{\sigma_{ij}^2}) \tag{9}$$

Previous research has shown that the propagation results highly depend on the kernel size σ_{ij} selection [22]. In [20], a method to learn the proper σ_{ij} for the kernel is introduced, which has excellent performance. However, it is not learnable in the case of few labeled data. In shape retrieval, since only the query shape has the label, the learning of σ_{ij} is not applicable. In our experiment, we use use an adaptive kernel size based on the mean distance to K-nearest neighborhoods [36]:

$$\sigma_{ij} = \alpha \cdot \text{mean}(\{knnd(x_i), knnd(x_j)\}) \tag{10}$$

where mean($\{knnd(x_i), knnd(x_j)\}$) represents the mean distance of the K-nearest neighbor distance of the sample x_i, x_j and α is an extra parameter. Both K and α are determined empirically.

VI. Shape Clustering based on the Learned distance

Besides the shape retrieval, the learned distance by the proposed approach could also be used for improving the performance of shape clustering. The difficulty of the shape clustering is also from the property of the shapes, which may have high variance in the same class and sometimes small difference in different classes. Similar to shape retrieval, the learned distance could improve the shape clustering results a lot. The basic idea here is using the learned distance to replace the original distance based on shape similarity, then the clustering algorithm can be applied on the new distance space.

In this paper, we choose Affinity Propagation [37] for shape clustering. Compared to other classic clustering algorithm, such as k-means, the main advantage of Affinity Propagation is that it doesn't need the prior knowledge for the number of clusters. As mentioned above, two shapes in the same class may be very different to each other and the distribution of difference is different for different classes. If the number of clusters is fixed before clustering, it may ruin the results because of the outliers. Therefore, Affinity Propagation is more suitable for the task of shape clustering, as the outliers or unusual shapes which are totally different to other shapes from the same class will be automatically classified to separate clusters and it will not affect other clusters. A simple review for Affinity Propagation algorithm is given below.

In [37], there are two kinds of messages communicated between data points: responsibility and availability, and each takes a different kind of competition into account.

To begin with, the availability are initialized to zero: $a(i, j) = 0$. The responsibility $r(i, j)$, sent from data point i to candidate exemplar point j, reflects the accumulated evidence for how well point j is to serve as the exemplar for point i, taking into account other potential exemplars for point i. The responsibilities are computed as

$$r(i, j) \longleftarrow s(i, j) - max_{j' \neq j}\{a(i, j') + s(i, j')\} \tag{11}$$

where $s(i, j)$ represents the similarity between data point j and i, which is directly obtained by $s(i, j) = -d(i, j)$, where $d(i, j)$ is the distance between data point j and i. For the learned

distance, as sim_t is the similarities between shapes, the similarity is transformed to distance by

$$D(x_i, x_j) = -\log(sim_t(x_i, x_j)) \tag{12}$$

since the shapes x_i, x_j are considered as the data points in the clustering algorithm, the distance $d(i,j) = D(x_i, x_j)$. For $j = i$, the self-responsibility $r(j,j)$ reflects accumulated evidence that point j is an exemplar, based on its input preference. In our experiments, $r(j,j)$ is determined by the average distance from data point j to all other data points.

The availability $a(i,j)$, sent from the candidate exemplar point j to point i, reflects the accumulated evidence for how appropriate it would be for point i to choose point j as its exemplar, taking into account the support from other points that point j should be an exemplar. Whereas the above responsibility update lets all candidate exemplars compete for ownership of a data point, the following availability update gathers evidence from data points as to whether each candidate exemplar would make a good exemplar:

$$a(i,j) \longleftarrow min\{0, r(j,j) + \sum_{i' \notin \{i,j\}} max\{0, r(i',j)\}\} \tag{13}$$

The self-availability $a(j,j)$ is updated differently:

$$a(j,j) \longleftarrow \sum_{i' \neq j} max\{0, r(i',j)\} \tag{14}$$

This message reflects accumulated evidence that point j is an exemplar, based on the positive responsibilities sent to candidate exemplar j from other points.

After the convergence, availability and responsibilities are combined to identify exemplars. For point i, its corresponding exemplar is obtained as

$$j^* = argmax_j\{a(i,j) + r(i,j)\} \tag{15}$$

This means to either identify point i as an exemplar if $j^* = i$, or identify data point j^* that is the exemplar for point i.

As shown in Section VII, the clustering results by Affinity Propagation based on the learned distance achieve a significant improvement on three challenge shape data sets than without learning.

VII. Experimental Results

In this section, we show that the proposed approach can significantly improve the performance of shape retrieval, shape classification and shape clustering of existing shape similarity methods.

A. Improving shape retrieval/matching

1) Improving MPEG-7 shape retrieval: The IDSC [3] significantly improved the performance of shape context [1] by replacing the Euclidean distance with shortest paths inside the shapes, and obtained the retrieval rate of 85.40% on the MPEG-7 data set. The proposed distance learning method is able to improve the IDSC retrieval rate to **91.00%**. For reference, Table I lists almost all the reported results on the MPEG-7 data set. The MPEG-7 data set consists of 1400 silhouette images grouped into 70 classes. Each class has 20 different shapes. The retrieval rate is measured by the so-called bull's eye score. Every shape in the database is compared to all other shapes, and the number of shapes from the same class among the 40 most similar shapes is reported. The bull's eye retrieval rate is the ratio of the total number of shapes from the same class to the highest possible number (which is 20×1400). Thus, the best possible rate is 100%. From the retrieval rates collected in Table I, we can clearly observe that our method made a big progress on this database, and the second highest result is 87.70% obtained by Shape Tree [7].

In order to visualize the gain in retrieval rates by our method as compared to IDSC, we plot the percentage of correct results among the first k most similar shapes in Fig. 4(a), i.e., we plot the percentage of the shapes from the same class among the first k-nearest neighbors for $k = 1, \ldots, 40$. Recall that each class has 20 shapes, which is why the curve increases for $k > 20$. We observe that the proposed method not only increases the bull's eye score, but also the ranking of the shapes for all $k = 1, \ldots, 40$.

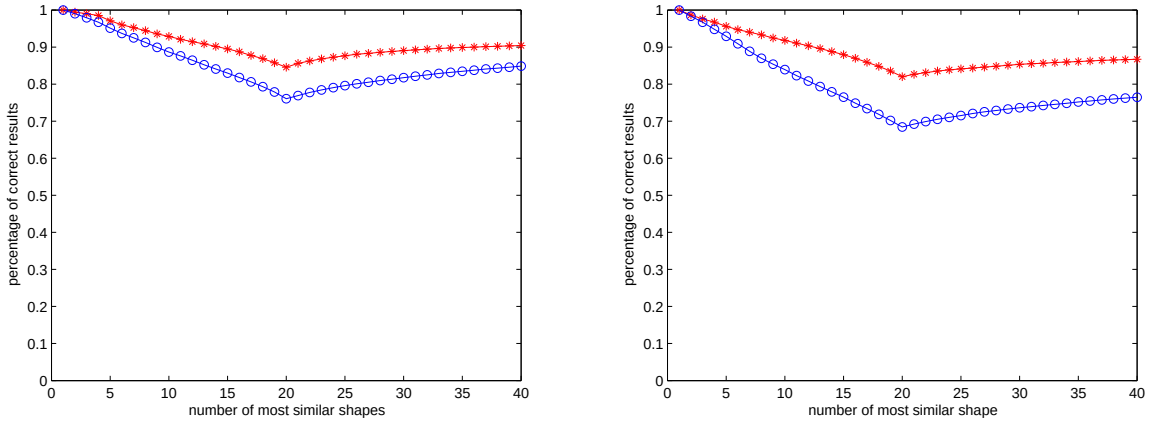
We use the following parameters to construct the affinity matrix: $\alpha = 0.25$ and the neighborhood size is $K = 10$. As stated in Section III, in order to increase computational efficiency, it is possible to construct the affinity matrix for only part of the database of known shapes. Hence, for each query shape, we first retrieve 300 the most similar shapes, and construct the affinity matrix W for only those shapes, i.e., W is of size 300×300 as opposed to a 1400×1400 matrix if we consider all MPEG-7 shapes. Then we calculate the new similarity measure sim_T for only those 300 shapes. Here we assume that all relevant shapes will be among the 300 most similar shapes. Thus, by using a larger affinity matrix we can improve the retrieval rate but at the cost of computational efficiency.

In addition to the statistics presented in Fig. 4, Fig. 5 illustrates also that the proposed approach improves the performance of IDSC. A very interesting case is shown in the first row, where for

TABLE I

RETRIEVAL RATES (BULL'S EYE) OF DIFFERENT METHODS ON THE MPEG-7 DATA SET.

Alg.	CSS [38]	Vis. Parts [4]	Shape Contexts [1]	Aligning Curves [39]	Distance Set [40]	Prob. Approach [41]	Chance Prob. [42]	Skeletal Context [43]	Gen. Model [2]	Optimized CSS [44]
Score	75.44%	76.45%	76.51%	78.16%	78.38%	79.19%	79.36%	79.92%	80.03%	81.12%
Alg.	Contour Seg. [45]	Shape L'Âne Rouge [46]	Multiscale Rep. [47]	Fixed Cor. [48]	Inner Distance [3]	Symbolic Rep. [27]	Hier. Procrustes [6]	Triangle Area [26]	Shape Tree [7]	IDSC [3] + our method
Score	84.33%	84.40%	84.93%	85.40%	85.40%	85.92%	86.35%	87.23%	87.70%	**91.00%**



(a) (b)

Fig. 4. (a) A comparison of retrieval rates between IDSC [3] (blue circles) and the proposed method (red stars) for MPEG-7. (b) A comparison of retrieval rates between visual parts in [4] (blue circles) and the proposed method (red stars) for MPEG-7.

IDSC only one result is correct for the query octopus. It instead retrieves nine apples as the most similar shapes. Since the query shape of the octopus is occluded, IDSC ranks it as more similar to an apple than to the octopus. In addition, since IDSC is invariant to rotation, it confuses the tentacles with the apple stem. Even in the case of only one correct shape, the proposed method learns that the difference between the apple stem is relevant, although the tentacles of the octopuses exhibit a significant variation in shape. We restate that this is possible because the new learned distances are induced by geodesic paths in the shape manifold spanned by the known shapes. Consequently, the learned distances retrieve nine correct shapes. The only wrong

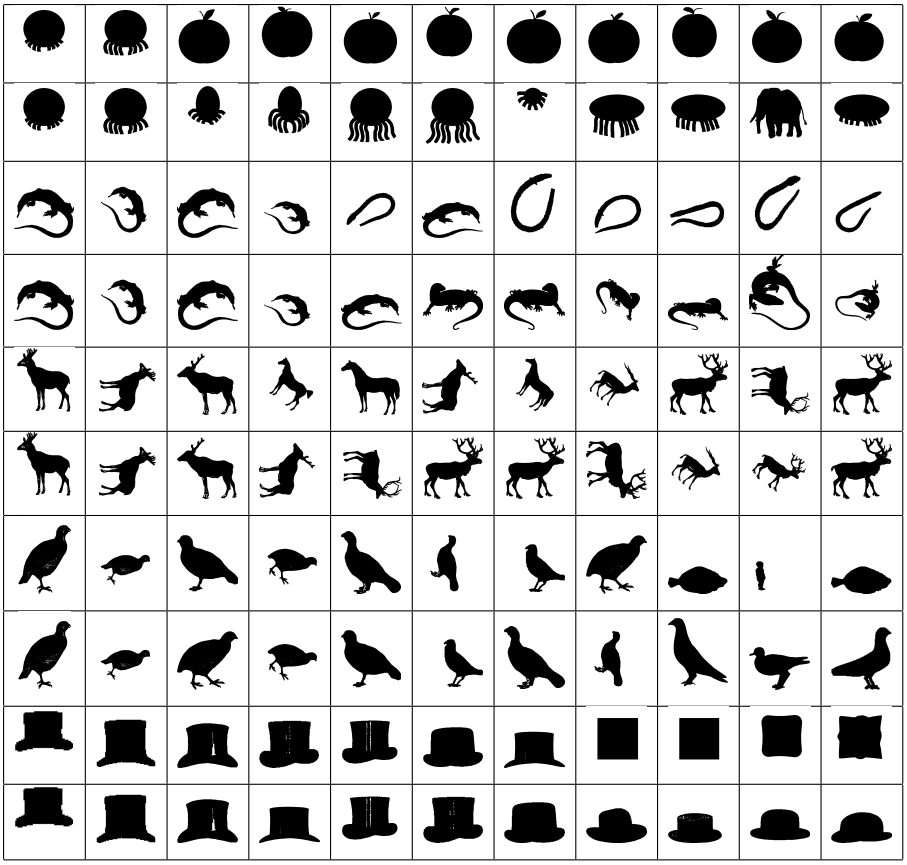
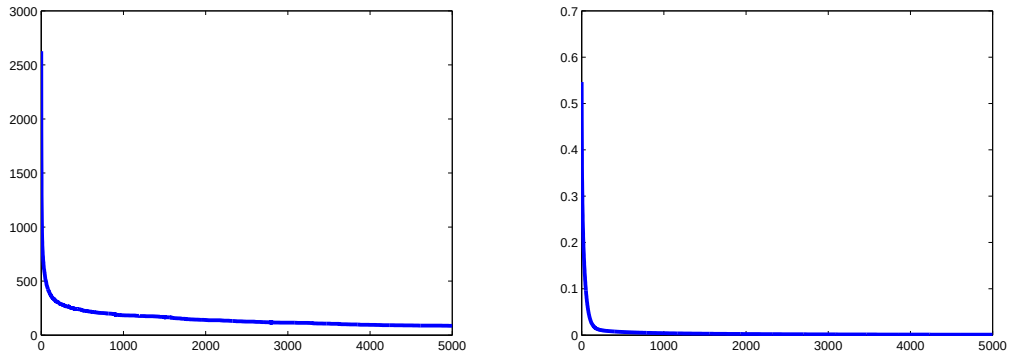


Fig. 5. The first column shows the query shape. The remaining 10 columns show the most similar shapes retrieved by IDSC (odd row numbers) and by our method (even row numbers).

results is the elephant, where the nose and legs are similar to the tentacles of the octopus.

As shown in the third row, six of the top ten IDSC retrieval results of lizard are wrong. since IDSC cannot ignore the irrelevant differences between lizards and sea snakes. All retrieval results are correct for the new learned distances, since the proposed method is able to learn the irrelevant differences between lizards and the relevant differences between lizards and sea snakes. For the results of deer (fifth row), three of the top ten retrieval results of IDSC are horses. Compared to it, the proposed method (sixth row) eliminates all of the wrong results so that only deers are in the top ten results. It appears to us that our new method learned to ignore the irrelevant small shape details of the antlers. Therefore, the presence of the antlers became a relevant shape feature here. The situation is similar for the bird and hat, with three and four wrong retrieval results respectively for IDSC, which are eliminated by the proposed method.



(a) (b)

Fig. 6. (a) The number of triangle inequality violations per iteration. (b) Plot of differences $||f_{t+1} - f_t||$ as a function of t.

An additional explanation of the learning mechanism of the proposed method is provided by examining the count of the number of violations of the triangle inequality that involve the query shape and the database shapes. In Fig. 6(a), the curve shows the number of triangle inequality violations after each iteration of our distance learning algorithm. The number of violations is reduced significantly after the first few hundred iterations. We cannot expect the number of violations to be reduced to zero, since cognitively motivated shape similarity may sometimes require triangle inequality violations [14]. Observe that the curve in Fig. 6(a) correlates with the plot of differences $||f_{t+1} - f_t||$ as a function of t shown in (b). In particular, both curves decrease very slow after about 1000 iterations, and at 5000 iterations they are nearly constant. Therefore, we selected $T = 5000$ as our stop condition. Since the situation is very similar in all our experiments, we always stop after $T = 5000$ iterations.

Besides the inner distance shape context [3], we also demonstrate that the proposed approach can improve the performance of **visual parts shape similarity** [4]. We select this method since it is based on very different approach than IDSC. In [4], in order to compute the similarity between shapes, first the best possible correspondence of visual parts is established (without explicitly computing the visual parts). Then, the similarity between corresponding parts is calculated and aggregated. The settings and parameters of our experiment are the same as for IDSC as reported in the previous section except we set $\alpha = 0.4$. The accuracy of this method has been increased from 76.45% to 86.69% on the MPEG-7 data set, which is more than 10%. This makes the

Fig. 7. Sample shapes from Kimia's 99 shape database [8]. We show two shapes for each of the 9 classes.

improved visual part method one of the top scoring methods in Table I. A detailed comparison of the retrieval accuracy is given in Fig. 4(b).

2) Improving Kimia's shape retrieval: Besides MPEG-7 database, we also present experimental results on the Kimia's 99 shape database [8]. The database contains 99 shapes grouped into nine classes. In this dataset, some images have protrusions or missing parts. Fig. 7 shows two sample shapes for each class of this dataset. As the database only contains 99 shapes, we calculate the affinity matrix based on all of the shape in the database. The parameters used to calculate the affinity matrix are: $\alpha = 0.25$ and the neighborhood size is $K = 4$. We changed the neighborhood size, since the data set is much smaller than the MPEG-7 data set. The retrieval results are summarized as the number of shapes from the same class among the first top 1 to 10 shapes (the best possible result for each of them is 99). Table II lists the numbers of correct matches of several methods. Again we observe that our approach could improve IDSC significantly, and it yields a nearly perfect retrieval rate, which is the best result in the Table II.

3) Improving Face Retrieval: We used a face data set from [49], where it is called *Face (all)*. It addresses a face recognition problem based on the shape of head profiles. It contains several head profiles extracted from side view photos of 14 subjects. There exist large variations in the shape of the face profile of each subject, which is the main reason why we select this data set. Each subject is making different face expressions, e.g., talking, yawning, smiling, frowning, laughing, etc. When the pictures of subjects were taken, they were also encouraged to look a little to the left or right, randomly. At least two subjects had glasses that they put on for half of their samples. A few sample pictures are shown in Fig. 8.

The head profiles are converted to sequences of curvature values, and normalized to the length of 131 points, starting from the neck area. The data set has two parts, training with 560 profiles and testing with 1690 profiles. The training set contains 40 profiles for each of the 14 classes.

TABLE II

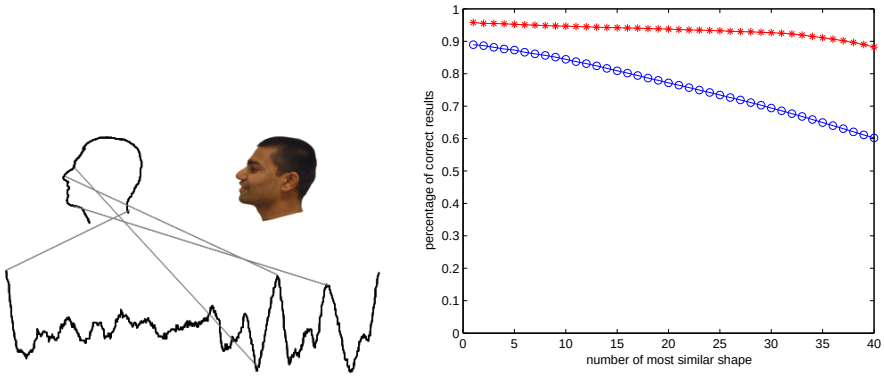
RETRIEVAL RESULTS ON KIMIA'S 99 SHAPE DATA SET [8]

Algorithm	1st	2nd	3rd	4th	5th	6th	7th	8th	9th	10th
SC [1]	97	91	88	85	84	77	75	66	56	37
Gen. Model [2]	99	97	99	98	96	96	94	83	75	48
Path Similarity [5]	99	99	99	99	96	97	95	93	89	73
Shock Edit [8]	99	99	99	98	98	97	96	95	93	82
IDSC [3]	99	99	99	98	98	97	97	98	94	79
Triangle Area [26]	99	99	99	98	98	97	98	95	93	80
Shape Tree [7]	99	99	99	99	99	99	99	97	93	86
Symbolic Rep. [27]	99	99	99	98	99	98	98	95	96	94
IDSC [3] + **our method**	99	99	99	99	99	99	99	99	97	99

Fig. 8. A few sample image of the *Face (all)* data set.

As reported on [49], we calculated the retrieval accuracy by matching the 1690 test shapes to the 560 training shapes. We used a dynamic time warping (DTW) algorithm with warping window [50] to generate the distance matrix, and obtained the 1NN retrieval accuracy of 88.9% By applying our distance learning method we increased the 1NN retrieval accuracy to 95.04%. The best reported result on [49] has the first nearest neighbor (1NN) retrieval accuracy of 80.8%. The retrieval rate, which represents the percentage of the shapes from the same class (profiles of the same subject) among the first k-nearest neighbors, is shown in Fig. 9(b).

The accuracy of the proposed approach is stable, although the accuracy of DTW decreases significantly when k increases. In particular, our retrieval rate for k = 40 remains high, 88.20%, while the DTW rate dropped to 60.18%. Thus, the learned distance allowed us to increase the retrieval rate by nearly 30%. Similar to the above experiments, the parameters for the affinity matrix is $\alpha = 0.4$ and $K = 5$.



(a) (b)

Fig. 9. (a) Conversion of the head profile to a curvature sequence. (b) Retrieval accuracy of DTW (blue circles) and the proposed method (red stars).

4) Improving leaf retrieval: The Swedish leaf data set comes from a leaf classification project at Linkoping University and Swedish Museum of Natural History [51]. Fig. 10 shows some representative examples. The data set contains isolated leaves from 15 different Swedish tree

Fig. 10. Typical images from the Swedish leaf database [51], one image per species. Note that some species are quite similar, e.g., the first, third and ninth species.

species, with 75 leaves per species. Same to the experimental method in Inner-distance Shape Contexts [3], 25 leaves of Each species are used for training, and the other 50 leaves are used for testing. The 1NN accuracy reported in [3] is 94.13%, but the results we obtained with their software[1] is 91.2%. Instead of the 1NN classification rate, we report the retrieval rate of the first 50 nearest neighbors. As the way we calculate the retrieval rate is similar to we did for MPEG-7 database, the Bull-eyes score, the score of the test is the ratio of the number of correct of all images to the highest possible number of hits(which is 25×750). Therefore, the retrieval rate

[1]http://vision.ucla.edu/∼hbling/code

will increase after the number of nearest neighbor is larger than 25. In Fig. 11, the retrieval rate of the Swedish leaf is improved a lot by the proposed approach, especially, the 1NN recognition rate is increased from 91.2% to 93.8%. Moreover,the parameters for the affinity matrix is $\alpha = 0.2$ and $K = 5$.

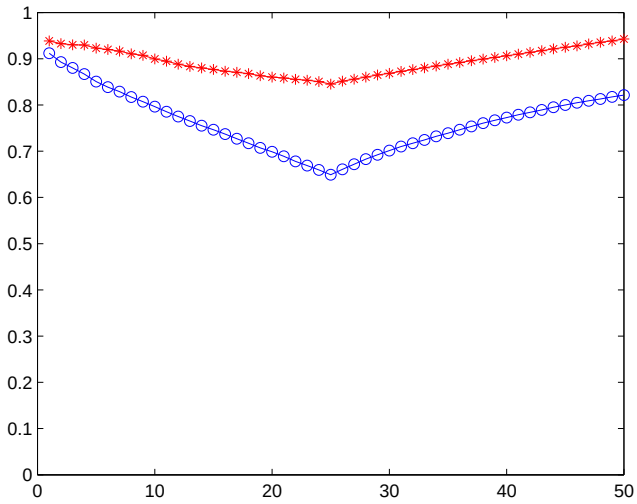


Fig. 11. Retrieval accuracy of Inner distance (blue circles) and the proposed method (red stars).

Fig. 12. Sample trademarks of our dataset used for the experiment

5) Improving trademark retrieval: With the increase of the registered trademarks, trademark retrieval is quite required by industry and commerce. Retrieval of trademark images by shape feature has proved a great challenge, though there has been considerable research into this topic

[52], [53]. Although Shape Contexts [1] did not give a full solution to shape-based trademark retrieval, we use it to calculate the shape similarity for trademark images and prove that our method has a potential to improve the retrieval results. A trademark dataset that contains 165 images are used for the experiment, which is collected by us from google. Some sample images of this dataset are shown in Fig. 12. For each image, we firstly computed its edge images with Canny operator [54], then randomly sample them into 300 points for matching with Shape Contexts. After we obtained the distance matrix, the proposed approach is used to learn the new distance, where the parameters are: $K = 5$ and $\alpha = 0.4$.

We only show several experimental results to demonstrate that our method can improve trademark retrieval. As shown in Fig. 13, the retrieval results by Shape Contexts and the improved retrieval results by the proposed method are listed together for comparison. We can clearly see the potential for our method to improve trademark retrieval.

B. Improving 1NN shape classification

The k-nearest neighbor algorithm is amongst the simplest of all machine learning algorithms. An object is classified by a majority vote of its neighbors, with the object being assigned to the class most common amongst its k nearest neighbors. k is a positive integer, typically small. If $k = 1$, then the object is simply assigned to the class of its nearest neighbor. The proposed distance learning algorithm could improve the recognition rate of 1NN classification. The retrieval results of *Face (all)* and Swedish leaf databases have shown the improvement. Besides, we divided the MPEG-7 dataset into two sets: training set and testing set. For each class, ten shapes are chosen as the training samples and the rest ten shapes are then used for testing. The results are shown in Table III. We can easily observe that all the performance on these datasets have been improved. The improvements on Swedish leaf and MPEG-7 are not so significant as on the Face dataset, which is normal due to two main reasons: 1) The 1NN classification rate on the Swedish leaf and MPEG-7 have already been very perfect with the original distances, which means there is no much potential on them; 2) The number of the training samples per class for the Swedish leaf and MPEG-7 are much fewer than the Face dataset. The parameters for all of the three datasets are the same to the retrieval setting.

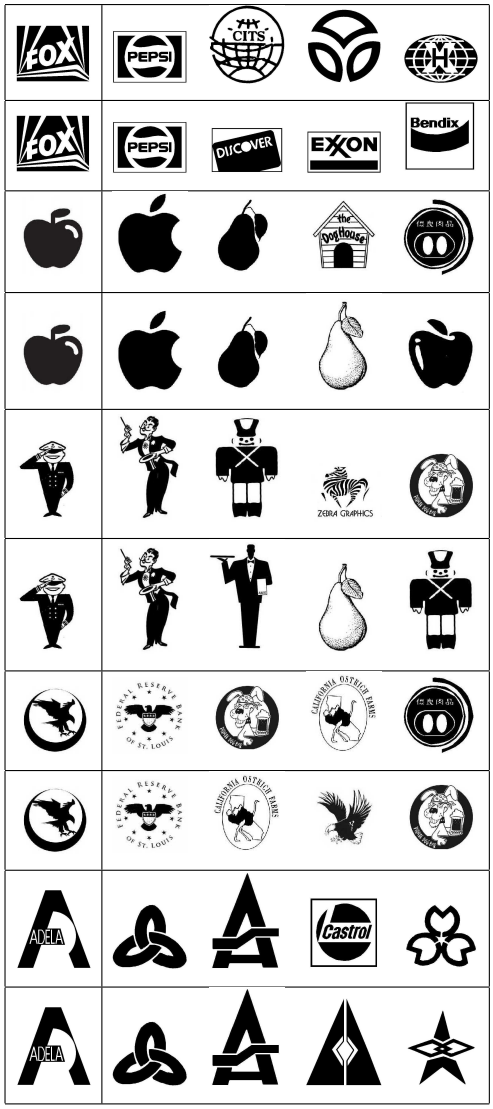


Fig. 13. The first column show the query trademark. The remaining 4 columns show the most similar trademarks retrieved by Shape Contexts [1] (odd row numbers) and by our method (even row numbers).

C. Improving Shape clustering

To evaluate the performance of the proposed approach on shape clustering in Section VI, we tried it on three standard databases: one is Kimia's 99 shape database [8] as shown in Fig. 7. Another one is Kimia's 216 shape database [8], which is a selected subset of MPEG-7 database. Fig. 14 shows two sample shapes for each class of Kimia's 216 shape database. The last database is the whole MPEG-7 database. In order to evaluate the performance of the shape clustering, an measurement is used to measure the accuracy of clustering, the accuracy of the pairwise

TABLE III

RESULTS OF 1NN CLASSIFICATION IMPROVEMENT

	Original Distance	Learned Distance
Face (all)	88.9%	95.4%
Swedish leaf	91.2%	93.8%
MPEG-7 database	94.7%	95.7%

correctly clustered objects. In other words, if two objects are clustered into one class and they are actually in the same class, this pair would be considered as correct. The score of the test is the ratio of the number of correct hits of all pairs of objects to the highest possible number of hits. The IDSC [3] is used to obtain the distance matrix of the three databases. The shape clustering results on both three databases based on the original distance by IDSC [3] and the learned distance by our algorithm proposed in Section VI are shown in Table VII-C. Notice that the learned distance achieved the significant improvement on all of the databases, and the numbers of the clusters are almost equal to the numbers of classes on Kimia's two databases. We believe that some other methods such as [12] can be also improved with our method. Here we did not compare with the shape clustering method in [12], since they need to fix the number of cluster centers before clustering.

As the goal of the shape clustering is different from the ranking, the number of iterations for the distance learning should be less than the ranking. The reason is there is only one labeled data point during the learning, if the number of iterations is too large, the difference between data points will be small, which cannot be used to distinguish the objects in different classes. For all of the clustering experiments, the number of iterations T is 1000 for MPEG-7 data set and 300 for the Kimia's two databases. The parameters to calculate the affinity matrix for MPEG-7 is the same to the retrieval. Besides, for Kimia's 99 shape database, the parameters are $K = 5$ and $\alpha = 0.33$, and for Kimia's 216 shape database, the parameters are $K = 7$ and $\alpha = 0.32$.

Fig. 14. Sample shapes from Kimia's 216 shape database [8]. We show two shapes for each of the 18 classes.

TABLE IV

RESULTS ON THE KIMIA'S 99 DATABASE [8], KIMIA'S 216 SHAPE DATABASE [8] AND MPEG-7 DATABASE.

	Kimia's 99 shape database		Kimia's 216 shape database		MPEG-7 shape database	
Number of Classes	9		18		70	
	Original Dist.	Learned Dist.	Original Dist.	Learned Dist.	Original Dist.	Learned Dist.
Number of Clusters	16	10	25	19	174	58
Accuracy	69%	95%	85%	97%	54%	86%

VIII. CONCLUSION AND DISCUSSION

In this work, we adapted a graph transductive learning framework to learn new distances with the application to shape retrieval, shape classification and shape clustering. The key idea is to replace the distances in the original distance space with distances induces by geodesic paths in the shape manifold. The merits of the proposed technique have been validated by significant performance gains over the experimental results. However, like semi-supervised learning, if there are too many outlier shapes in the shape database, the proposed approach cannot improve the results. Our future work will focus on addressing this problem. We also observe that our method is not limited to 2D shape similarity but can also be applied to 3D model retrieval, which will also be part of our future work.

ACKNOWLEDGEMENTS

We would like to thank Haibin Ling for proving us his software for Inner-Distance method and the Swedish leaf database. We would like to thank Eamonn Keogh for providing us the *Face (all)* dataset. We also want to thank B.B. Kimia for proving his shape databases on the internet.

This work was support in part by the NSF Grant No. IIS-0534929 and by the DOE Grant No. DE-FG52-06NA27508.

REFERENCES

[1] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *IEEE Trans. PAMI*, vol. 24, pp. 705–522, 2002.

[2] Z. Tu and A. L. Yuille, "Shape matching and recognition - using generative models and informative features," in *ECCV*, 2004, pp. 195–209.

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

[4] L. J. Latecki and R. Lakämper, "Shape similarity measure based on correspondence of visual parts," *IEEE Trans. PAMI*, vol. 22, no. 10, pp. 1185–1190, 2000.

[5] X. Bai and L. J. Latecki, "Path similarity skeleton graph matching," *IEEE Trans. PAMI*, vol. 30, no. 7, pp. 1282–1292, 2008.

[6] G. McNeill and S. Vijayakumar, "Hierarchical procrustes matching for shape retrieval," in *Proc. CVPR*, 2006.

[7] P. F. Felzenszwalb and J. Schwartz, "Hierarchical matching of deformable shapes." in *CVPR*, 2007.

[8] T. B. Sebastian, P. N. Klein, and B. B. Kimia, "Recognition of shapes by editing their shock graphs," *IEEE Trans. PAMI*, vol. 25, pp. 116–125, 2004.

[9] K. Siddiqi, A. Shokoufandeh, S. J. Dickinson, and S. W. Zucker, "Shock graphs and shape matching," *Int. J. of Computer Vision*, vol. 35, pp. 13–32, 1999.

[10] A. Shokoufandeh, D. Macrini, S. Dickinson, K. Siddiqi, and S. W. Zucker, "Indexing hierarchical structures using graph spectra," *IEEE Trans. PAMI*, vol. 27, no. 7, pp. 1125–1140, 2005.

[11] L. Gorelick, M. Galun, E. Sharon, R. Basri, and A. Brandt, "Shape representation and classification using the poisson equation," *IEEE Trans. PAMI*, vol. 28, no. 12, pp. 1991–2005, 2006.

[12] A. Srivastava, S. H. Joshi, W. Mio, and X. Liu, "Statistic shape analysis: clustering, learning, and testing," *IEEE Trans. PAMI*, vol. 27, pp. 590–602, 2005.

[13] X. Zhu, "Semi-supervised learning with graphs," in *Doctoral Dissertation*, 2005, pp. Carnegie Mellon University, CMU–LTI–05–192.

[14] J. Vleugels and R. Veltkamp, "Efficient image retrieval through vantage objects," *Pattern Recognition*, vol. 35 (1), pp. 69–80, 2002.

[15] L. J. Latecki, R. Lakämper, and U. Eckhardt, "Shape descriptors for non-rigid shapes with a single closed contour," in *CVPR*, 2000, pp. 424–429.

[16] X. Yang, X. Bai, L. J. Latecki, and Z. Tu, "Improving shape retrieval by learning graph transduction," in *ECCV*, 2008.

[17] U. Brefeld, C. Buscher, and T. Scheffer, "Multiview dicriminative sequential learning," in *ECML*, 2005.

[18] N. D. Lawrence and M. I. Jordan, "Semi-supervised learning via gaussian processes," in *NIPS*, 2004.

[19] T. Joachims, "Transductive inference for text classification using support vector machines," in *ICML*, 1999, pp. 200–209.

[20] X. Zhu, Z. Ghahramani, and J. Lafferty., "Semi-supervised learning using gaussian fields and harmonic functions," in *ICML*, 2003.

[21] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Scholkopf., "Learning with local and global consistency," in *NIPS*, 2003.

[22] F. Wang, J. Wang, C. Zhang, and H. Shen., "Semi-supervised classification using linear neighborhood propagation," in *CVPR*, 2006.

[23] D. Zhou, J. Weston, A.Gretton, Q.Bousquet, and B.Scholkopf., "Ranking on data manifolds," in *NIPS*, 2003.

[24] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, pp. 2323–2326, 2000.

[25] X. Fan, C. Qi, D. Liang, and H. Huang, "Probabilistic contour extraction using hierarchical shape representation," in *Proc. ICCV*, 2005, pp. 302–308.

[26] N. Alajlan, M. Kamel, and G. Freeman, "Geometry-based image retrieval in binary image databases," *IEEE Trans. on PAMI*, vol. 30, no. 6, pp. 1003–1013, 2008.

[27] M. Daliri and V. Torre, "Robust symbolic representation for shape recognition and retrieval," *Pattern Recognition*, vol. 41, no. 5, pp. 1799–1815, 2008.

[28] J. Yu, J. Amores, N. Sebe, P. Radeva, and Q. Tian, "Distance learning for similarity estimation," *IEEE Trans. PAMI*, vol. 30, pp. 451–462, 2008.

[29] E. Xing, A. Ng, M. Jordanand, and S. Russell, "Distance metric learning with application to clustering with side-information," in *NIPS*, 2003, pp. 505–512.

[30] A. Bar-Hillel, T. Hertz, N. Shental, and D. Weinshall, "Learning distance functions using equivalence relations," in *ICML*, 2003, pp. 11–18.

[31] V. Athitsos, J. Alon, S. Sclaroff, and G. Kollios, "Bootmap: A method for efficient approximate similarity rankings," in *CVPR*, 2004.

[32] T. Hertz, A. Bar-Hillel, and D. Weinshall, "Learning distance functions for image retrieval," in *CVPR*, 2004, pp. 570–577.

[33] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: Bringing order to the web," *Stanford Digital Libraries Working Paper*, 1998.

[34] L. Zelnik-Manor and P. Perona, "Self-tuning spectral clustering," in *NIPS*, 2004.

[35] M. Hein and M. Maier, "Manifold denoising," in *NIPS*, 2006.

[36] 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.

[37] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *Science*, vol. 315, pp. 972–976, 2007.

[38] F. Mokhtarian, F. Abbasi, and J. Kittler, "Efficient and robust retrieval by shape content through curvature scale space," *Image Databases and Multi-Media Search, A.W.M Smeulders and R. Jain eds*, pp. 51–58, 1997.

[39] T. Sebastian, P. Klein, and B. Kimia, "On aligning curves," *IEEE Trans. PAMI*, vol. 25, pp. 116–125, 2003.

[40] C. Grigorescu and N. Petkov, "Distance sets for shape filters and shape recognition," *IEEE Trans. on Image Processing*, vol. 12, no. 7, pp. 729–739, 2003.

[41] G. McNeill and S. Vijayakumar, "2d shape classification and retrieval," in *IJCAI*, 2005.

[42] B. Super, "Learning chance probability functions for shape retrieval or classification," in *Proceedings of the IEEE Workshop on Learning in CVPR*, 2004.

[43] J. Xie, P. Heng, and M. Shah, "Shape matching and modeling using skeletal context," *Pattern Recognition*, vol. 41, no. 5, pp. 1756–1767, 2008.

[44] F. Mokhtarian and M. Bober, *Curvature Scale Space Representation: Theory, Applications & MPEG-7 Standardization*. Dordrecht: Kluwer Academic Publishers, 2003.

[45] E. Attalla and P. Siy, “Robust shape similarity retrieval based on contour segmentation polygonal multiresolution and elastic matching,” *Pattern Recognition*, vol. 38, no. 12, pp. 2229–2241, 2005.

[46] A. Peter, A. Rangarajan, and J. Ho, “Shape l’âne rouge: Sliding wavelets for indexing and retrieval,” in *CVPR*, 2008.

[47] T. Adamek and N. O’Connor, “A multiscale representation method for nonrigid shapes with a single closed contour,” *IEEE Trans. on CSVT*, vol. 14, no. 5, pp. 742–753, 2004.

[48] B. Super, “Retrieval from shape databases using chance probability functions and fixed correspondence,” *Int. J. Pattern Recognition Artif. Intell.*, vol. 20, no. 8, pp. 1117–1137, 2006.

[49] E. Keogh, “UCR time series classification/clustering page,” in *http://www.cs.ucr.edu/˜ eamonn/time_series_data/*.

[50] C. A. Ratanamahatana and E. Keogh, “Three myths about dynamic time warping,” in *SDM*, 2005, pp. 506–510.

[51] O. Soderkvist, *Computer vision classification of leaves from swedish trees*. Master’s thesis, Linkoping University, 2001.

[52] A. K. Jain and A. Vailaya, “Shape-based retrieval: A case study with trademark image databases,” *Pattern Recognition*, vol. 31, no. 9, pp. 1369–1390, 1998.

[53] Y. S. Kim and W. Y. Kim, “Content-based trademark retrieval system using a visually salient feature,” *Image and Vision Computing*, vol. 16, no. 12–13, pp. 931–939, 1998.

[54] J. Canny, “A computational approach to edge detection,” *IEEE Trans. PAMI*, vol. 8, no. 6, pp. 679–698, 1986.