Shape Similarity Analysis by Self-Tuning Locally Constrained Mixed-Diffusion

Lei Luo, Chunhua Shen, Chunyuan Zhang, *Member, IEEE*, Anton van den Hengel, *Member, IEEE*

***Abstract*—Similarity analysis is a powerful tool for shape matching/retrieval and other computer vision tasks. In the literature, various shape (dis)similarity measures have been introduced. Different measures specialize on different aspects of the data. In this paper, we consider the problem of improving retrieval accuracy by systematically fusing several different measures. To this end, we propose the locally constrained mixed-diffusion method, which partly fuses the given measures into one and propagates on the resulted locally dense data space. Furthermore, we advocate the use of self-adaptive neighborhoods to automatically determine the appropriate size of the neighborhoods in the diffusion process, with which the retrieval performance is comparable to the best manually tuned kNNs. The superiority of our approach is empirically demonstrated on both shape and image datasets. Our approach achieves a score of 100% in the bull's eye test on the MPEG-7 shape dataset, which is the best reported result to date.**

***Index Terms*—Shape similarity analysis, shape/image retrieval, locally constrained mixed-diffusion.**

I. Introduction

Shape retrieval tasks require that we recover from a database a set of shapes which are most similar to a query shape. An important component of most retrieval system is the distance measure used to measure shape similarity. A wide range of studies have focused on designing robust and informative shape representations in order to achieve high retrieval accuracy [1]–[5]. Each representation tends to emphasize a different aspect of object shape, however, and thus no definitive solution has been identified. Overcoming this problem requires reconciling shape information from different parts of an object, or different scales of analysis. The broader structural similarity between shapes thus needs to be reconciled with the differences in fine-grained surface properties, for instance, and the influence of potentially identical parts reconciled against largely dissimilar wholes. It is unrealistic to design a universal representation that works well for all problems on all datasets. Moreover, the pairwise similarity or distance defined on most conventional representations is often incapable of capturing category-level information across classes. To exploit these high-level relationships, it requires more sophisticated analysis of shapes.

Recently, many learning-based methods have been proposed to conquer the two above-mentioned problems in shape retrieval. These methods exploit the underlying structure of the datasets to improve the retrieval accuracy obtained by existing ranking approaches [6]–[11]. Other approaches such as rank aggregation concentrate on integrating different rankings into a single more accurate measure [12]–[15].

In this work, we consider the problem of improving retrieval accuracy by fusing different similarity or distance measures. Because different measures specialize on different aspects of the data, the intuition here is that better performance might be achieved by combining multiple complementary measures. Our goal is that in an integrated ranking system, an instance being ranked high in all measures should be at the top. At the same time, an instance being ranked high in a particular measure also deserves a position in the retrieved list.

If we consider, for example, Shape Contexts (SC) [2] and Inner-Distance Shape Contexts (IDSC) [1], which are two state-of-the-art shape distance measures. SC encodes primarily global shape information and generally works very well for rigid objects. IDSC focuses instead on internal structures and is thus more effective for articulated and deformable shapes. Given a query shape q, let A_q and B_q be the set of database shapes ranked highly by SC and IDSC respectively. A shape x may be relevant to q in some cases while $x \in A_q \cup B_q$, but the probability is higher when $x \in A_q \cap B_q$. Building upon this fact, the proposed mixed-diffusion method integrates both global and local information into a measure which is capable of effectively using information from both measures to achieve a more accurate and robust result.

Fig. 1 shows 9 shapes returned when querying a shape from the MPEG-7 dataset [16] as measured by SC, IDSC and our method. The methods return 13, 12 and 20 shapes respectively of the 20 correct shapes in the "bull's eye test". Besides retrieving better matched shapes, our method also improves the ranking order of the results. For example, the ninth result of SC is ranked higher using IDSC. Our method returns the object in the third position, which better reflects its visual similarity to the query.

The main idea of our proposed Locally Constrained Mixed-Diffusion (LCMD) method is to fuse a few given measures into a single one and then propagate on it using the Locally Constrained Diffusion Process (LCDP) of [8]. A straightforward fusion approach is to linearly combine measures altogether, which often submerges useful information in background. On the other hand, as a promising diffusion method, LCDP performs well only in a dense data space. In [8], the authors *densify* the data space by adding so-called "ghost points"

L. Luo and C. Zhang are with the College of Computer, National University of Defense Technology, Changsha, Hunan, 410073, China. L. Luo's contribution was made when he was a visiting student at the Australian Centre for Visual Technologies, The University of Adelaide. (e-mail: {l.luo, cyzhang}@nudt.edu.cn).

C. Shen and A. van den Hengel are with the Australian Centre for Visual Technologies, and the School of Computer Science, The University of Adelaide, Adelaide, SA 5005, Australia (e-mail: {chunhua.shen, anton.vandenhengel}@adelaide.edu.au). Correspondence should be addressed to C. Shen.

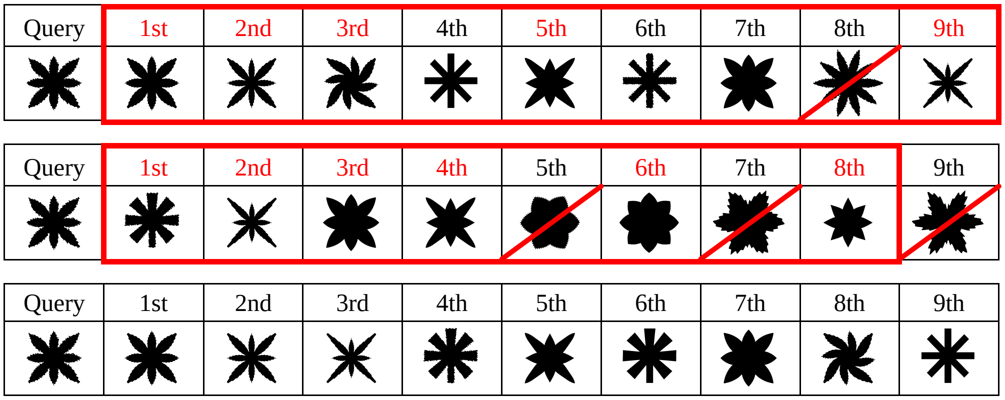


Fig. 1. **Top to bottom**: The retrieval results of SC, IDSC and LCMD+SAN on the MPEG-7 shape dataset [16] using a query from the MPEG-7 dataset. Incorrect results (according to [16]) are marked with slash, and shapes with red ordinal numbers and shapes in red boxes are DNs and SANs of the query, respectively.

(GP). However, as we show later, unsupervised GP gains little in our experiments (see Fig. 3(a) for example). In LCDM, we set the affinity scores of distant instances to zeros in all measures, and then linearly combine them together for the next diffusion process. Therefore, the useful information of each measure is kept in diffusion because most of irrelevant instances are excluded. Furthermore, the reserved neighbors in different measures form a locally dense space, which meets the requirement of LCDP.

Closest work to ours is Bai *et al.* [12]. Their co-transduction method fuses different measures through a semi-supervised learning framework. However, the co-transduction method only makes use of $A_q \cup B_q$ to retrieval more in the SC and IDSC example. The ranking order of an element in the fusing result is less considered. In contrast, we not only reserve neighbors to reveal $A_q \cup B_q$, but also combine them to utilize $A_q \cap B_q$. Objects occurring both in A_q and B_q achieve higher positions in our results. Besides, the performance of co-transduction is highly dependent on the original measures' first-ranking accuracies. LCDP is robust to noise in that it works on a neighborhood graph of the data. By exploiting the robustness of LCDP, the proposed method reduces the dependence.

The size of neighborhood has important impact on the performance of LCDP. Ideally, the neighborhood size should be data dependent. Yang *et al.* [6] proposed the Dominant Neighborhood (DN) to automatically determine the optimal number of neighbors, which reduces the risk of using sub-optimal neighbors in LCDP. However, the implementation of DN still needs to specify initial k-nearest neighbors (kNN) for each dataset. As demonstrated in Fig. 1, shapes marked with red ordinal numbers are dominant neighbors (DN's) of the query. The pattern consisting of the 4th and 6th shapes in the first row is lost, while the incorrect 5th and 7th shapes in the second row are excluded. We extend DN to Self-Adaptive Neighborhood (SAN) in two folds. First, we weight the neighbors by their affinities against the query, which helps determine the size of neighborhood. Second, we reserve other high rankings besides dominant neighbors, which helps preserve diverse patterns of the same class. The self-adaptive neighbors of the query are marked with red boxes in Fig. 1. Although including some incorrect neighbors, the LCMD process can complement and benefit from the diverse patterns of SAN. Briefly, the greatest advantage of SAN is its self-adaptation. It cannot perform better than the best manually set kNN, although the results are acceptable as shown in our experiments. In the next section, we briefly review some work that is closest to ours. Then the locally constrained mixed-diffusion method and the self-adaptive neighborhood are introduced in Sec. III and Sec. IV, respectively. Sec. V describes how to construct the affinity matrices from the given distance measures. Sec. VI shows the experimental results on both shape and image datasets.

II. Related work

Since a large number of shape similarity methods have been proposed in the literature, here we only focus on some recent methods that are relevant to ours. Shape Context (SC) introduced by Belongie *et al.* [2] might be one of most popular shape descriptors. SC encodes the global information of a shape into a histogram and generally works very well for rigid objects. Ling and Jacobs [1] introduced Inner-Distance Shape Context (IDSC) by replacing the Euclidean distance used in SC with the geodesic distance, which is more suitable for articulated shapes. Gorelick *et al.* [4] divided a shape into parts by describing shapes using Poisson equations. Felzenszwalb and Schwartz [3] instead decomposed the boundary into segments and represented a shape hierarchically. Gopalan *et al.* [5] proposed to combine shape decomposition and IDSC to cope with non-planar shapes. Their Articulation-Invariant Representation (AIR) approach obtains the best retrieval rate among non-learning based methods in the "bull's eye test" on the MPEG-7 dataset. However, the pairwise similarity defined on these representations is often incapable of capturing category-level information across classes. Recently, many learning-based methods have been proposed to exploit the high-level relationships, which can loosely be divided into two categories—post-processing and rank aggregation.

Post-processing methods usually take an initial ranking based on an existing approach and then exploit the underlying structure to improve the results. Yang *et al.* [10] proposed to improve shape retrieval accuracy by exploiting the underlying shape manifold structure. They learned the similarity of two shapes from their context information with Label Propagation (LP). Kontschieder *et al.* [9] proposed a modified mutual kNN graph to represent the shape manifold. Yang *et al.* [8] added synthetic points to *densify* the shape space due to the fact that the diffusion process may not propagate properly in a spare graph space. They also proposed LCDP to improve the noise stability, which we have borrowed in our method. Jegou *et al.* [17] introduced the Contextual Dissimilarity Measure (CMD) which takes the neighborhood of an image into account. Pedronette and Torres [7] introduced a similar contextual space for image re-ranking. Recently, Yang and Latecki [6] proposed to learn on Tensor Product Graphs (TPG) to better reveal the intrinsic structure of the data manifold, which achieved promising results on the MPEG-7 shape dataset.

Rank aggregation methods fuse multiple measurements to improve accuracy in retrieval tasks. Santini and Jain [14] proposed the fuzzy feature contrast model to integrate several features based on fuzzy logic, which has been applied successfully in an content-based image retrieval system [18]. Zhou and Burges [13] learned multiple views of the same instance with a Markov mixture model. In their method, each view of the data is represented as a graph. Bai *et al.* [12] considered the query as the only labeled instance and then fused different similarity measures within the co-training [19] framework. We implement rank aggregation in a novel way.

Many of the above-mentioned methods need to use kNN to convert distances into similarities [6], [8]–[10], [12], or to restrict the diffusion process [6], [8], [9], [17]. In general, the size of kNN is critical and difficult to determine. The common practice has been trial-and-error for k on different datasets, or for different instances on the same dataset. In [6], the authors introduced the notion of Dominant Neighborhood (DN) to capture the most supportive neighbors of kNN. Elements that are not strongly associated with others in the kNN are excluded from post-processing, which mitigates the risk of using wrong neighbors, yet it may lose some patterns of the class to which the query belongs. Moreover, the implementation of DN still has to specify k for different datasets. Here we extend DN such that the size of the neighborhood can be adaptively determined without using heuristics.

Next we present our main results.

III. Locally constrained mixed-diffusion

Given a set of data points $X = \{x_1, ...x_n\}$ and several different similarity measures $\{S_1, ..., S_m\}$, where the similarity of x_i and x_j in S_l is defined as $\rho_l(x_i, x_j)$. For each measure S_l, there is a fully connected directed graph $G_l = (V, E, w_l)$, which shares the same sets of vertexes $V = X$ and edges $E = \{E(i,j) | E(i,j) = \langle x_i, x_j \rangle, \forall x_i, x_j \in X, i \neq j\}$. The edge of G_l is labeled with the strength of affinity $w_l(i,j) = \rho_l(x_i, x_j)$. Now the challenge is how to fuse $G_1, ..., G_m$ altogether to utilize the abundant information of multi-measures. A simple approach is to linearly combine them:

$$w(i,j) = \sum\nolimits_{l=1}^{m} c_l w_l(x_i, x_j) = \sum\nolimits_{l=1}^{m} c_l \rho_l(x_i, x_j), \quad (1)$$

where $w(i,j)$ is the weight of $E(i,j)$ on the fused graph $G = (V, E, w)$, $c_l \in [0,1]$ denotes the contribution of each measure and we have $\sum_{l=1}^{m} c_l = 1$. The weight c_l deserves a large value if the corresponding graph S_l is reliable. G represents a Markov chain on X, whose transition matrix $P = [p_{ij}]_{n \times n}$, with

$$p_{ij} = \frac{w(i,j)}{\sum_{j=1}^{n} w(i,j)}. \quad (2)$$

p_{ij} represents the probability of transition in one time step from node x_i to node x_j, and $P^{(t)}$—the t-th power of P—gives the transition probability in t time steps. Then, a graph diffusion procedure can be performed on G to exploit the underlying relation between the data points [20].

Here a potential problem is that useful information of some measures may be submerged into the background during the construction of G, which may weaken the underlying relation and make the diffusion process fail. Let us consider the following extreme example. Assume $c_1 = ... = c_m = \frac{1}{m}$; x_i and x_j are immediate neighbors in G_p with a large weight $w_p(i,j) = Cm\mu$ while they are far away in other measures with $w_l(i,j) < \mu$ $(l = 1, ..., p-1, p+1, ..., m)$; x_k is a far neighbor of x_i with $w_l(i,k) = (C+1)\mu$ in all measures. Here μ is the mean weight of the linearly combined graph G and C is a large positive constant. Then $w(i,j) = \sum_{l=1}^{m} c_l w_l(x_i, x_j) = \frac{1}{m} \sum_{l=1}^{m} w_l(x_i, x_j) < C\mu + \frac{m-1}{m}\mu < (C+1)\mu = w(i,k)$. Thus, the weight between x_j and x_i is smaller than the weight between x_k and x_i—the information of the significantly large affinity $w_p(i,j)$ is lost in the background. In general, two possible cases can lead to a large $w_p(i,j)$: noise, which will be discussed later; or a strong similarity between x_i and x_j in a particular measure of G_p. For the second possibility, as discussed in the above extreme example, $w_p(i,j)$ can be submerged into the background and this incurs an irreversible loss. Usually, one cares more about a salient feature in one measure than a trivial feature appearing in multiple measures. But a diffusion process built on a misleading graph (e.g., the graph obtained by naive linear combination of multiple graphs in (2)) is doomed to fail to capture the underlying data structure that one is really interested in. In [13], a Markov mixture model was proposed, which is able to keep the information of all measures of the data. However, salient information in a particular measure may still be submerged into the background.

In this work, we propose to preserve the useful information using LCMD. To this end, we first simplify the fully-connected graphs $G_1, ..., G_m$ using kNN graphs $G_{1K}, ..., G_{mK}$, where $w_{lK}(i,j) = w_l(i,j) = \rho_l(i,j)$ if x_j belongs to kNN of x_i on G_l and $w_{lK}(i,j) = 0$ otherwise. Then we fuse them altogether to G_K. If there is an edge between x_i and x_j on G_{lK}, there is an edge on G_K, whose weight is defined as:

$$w_K(i,j) = \sum\nolimits_{l=1}^{m} y_l c_l \rho_l(x_i, x_j), \quad (3)$$

where $y_l = 1$ if $w_{lK} > 0$ and $y_l = 0$ if $w_{lK} = 0$. Thus, $w_K(i,j)$ is not zero only if x_j is a kNN of x_i in at least one measure, and at most mkn edges are kept in G_K. Since the edges with small weights are all set to zeros, a strong edge in a particular measure contributes more in G_K. In the previous example, although the weight $w_K(i,j) = \sum_{l=1}^{m} y_l c_l \rho_l(x_i, x_j) = c_p \rho_p(x_i, x_j) = C\mu$ is small, it is larger than at least $\binom{n}{2} - mkn$ zero-weight edges.

Then, the transition probability of the Markov chain on G_K is $P_K = [p_{ijK}]_{n \times n}$, where:

$$p_{ijK} = \frac{w_K(i,j)}{\sum_{j=1}^{n} w_K(i,j)}. \tag{4}$$

We can run the chain forward in time, which means taking larger power of P_K. Note that $P_K^{(t)}$ reveals the relevant local geometric structures of G_K at the scale of t. This simple Markov based diffusion strategy can be problematic because the relation between x_i and x_j depends on the narrow connection of kNN(x_i) and kNN(x_j) while the reserved edges in G_K may sometimes be caused by noise in some particular measures. We propose to employ LCDP [8] to exclude the influence of noise and outliers in the diffusion process. LCDP extends the connection by considering the paths between kNNs of x_i and kNNs of x_j. The diffusion process can then be defined as:

$$P_{KK}^{(t+1)} = P_K^T P_{KK}^{(t)} P_K, \tag{5}$$

where $P_{KK}^{(0)} = P$ is the original transition probability matrix computed by (2). If most paths from kNNs of x_i to kNNs of x_j are short, the two kNN are compatible and therefore the probability of transition from x_i to x_j is high. Meanwhile, the influence of noise and outliers in the diffusion process is reduced since more paths are considered in diffusion. On the other hand, more paths means that more data are required. As a result, LCDP works well only in a dense data space. A kNN graph constructed from a single measure may not provide a sufficiently dense space. As shown in [8], one can densify it by adding GPs. However, compared with supervised-GP, unsupervised-GP gains little in the experiments as discussed in [8] (also in Table I of our experiments). LCMD avoids the supervised procedure as the reserved neighbors in different measures form a locally dense space in G_K.

IV. Self-adaptive neighborhood

The size of neighborhood k is critical for defining the local structure. The original LCDP needs to manually tune k for different datasets. Even worse, ideally, the appropriate size should even be different for different instances of the same dataset. As a global parameter, setting a single k is unlikely to capture the local structures of the data, especially when the sizes of classes are significantly different.

Yang and Latecki [6] introduced DN based on the dominant set [21] to automatically determine the optimal number of neighbors. A dominant set is a subset of the data which corresponds to a maximally cohesive cluster. The dominant neighborhood DN(x_i) of a data point x_i is the dominant set of kNN(x_i). DN(x_i) reserves the most compact neighborhood structure of kNN(x_i) while excluding the points that are not strongly associated with others. However, it still requires to manually specify the value for k. Furthermore, some neighbors belonging to other patterns of the same class of x_i may be discarded since they are highly different from DN(x_i).

To better reveal the neighborhood structure of each data point, we propose a novel method, termed Self-Adaptive Neighborhood (SAN). Instead of directly using DN(x_i), we improve it using the classic ϵ-nearest-neighbor (ϵ-NN) theory. Assuming that $x_{f(i)}$ is the farthest neighbor of x_i in DN(x_i), the affinity between x_i and $x_{f(i)}$ serves as the radius of ϵ-NN in SAN. Consequently, as shown in the red boxes of Fig. 1, besides the dominant neighbors marked with red ordinal numbers, other neighbors of x_i still have a chance to contribute in the next diffusion process, which complements the single pattern of DN.

Let $W = [w_{ij}]_{n \times n}$ be the affinity matrix computed by an original measure S. We assume that W is symmetric (if not, replacing W by $\frac{W+W^T}{2}$). Then, an indicator vector $\mathbf{z}_i = (z_{i1}, ..., z_{in})$ is introduced for each data point x_i. The dominant set DN(x_i) of kNN(x_i) can be obtained by solving the following quadratic program [6]:

$$\begin{aligned} \max_{\mathbf{z}_i} \quad & f(\mathbf{z}_i) = \mathbf{z}_i^T W \mathbf{z}_i \\ & z_{ij} = 0, x_j \notin k\text{NN}(x_i); \\ \text{s.t.} \quad & \sum_{j=1}^{n} z_{ij} = 1; \\ & z_{ij} \geq 0, j = 1, ..., n. \end{aligned} \tag{6}$$

Each neighbor in kNN(x_i) is treated equally in the DN method. So, when the selected size of kNN(x_i) is significantly larger than the intrinsic size of x_i's neighborhood, the resulting DN(x_i) may be a compact cluster from a different class. This explains why the work of [6] has to specify k for different datasets in order to achieve good performance.

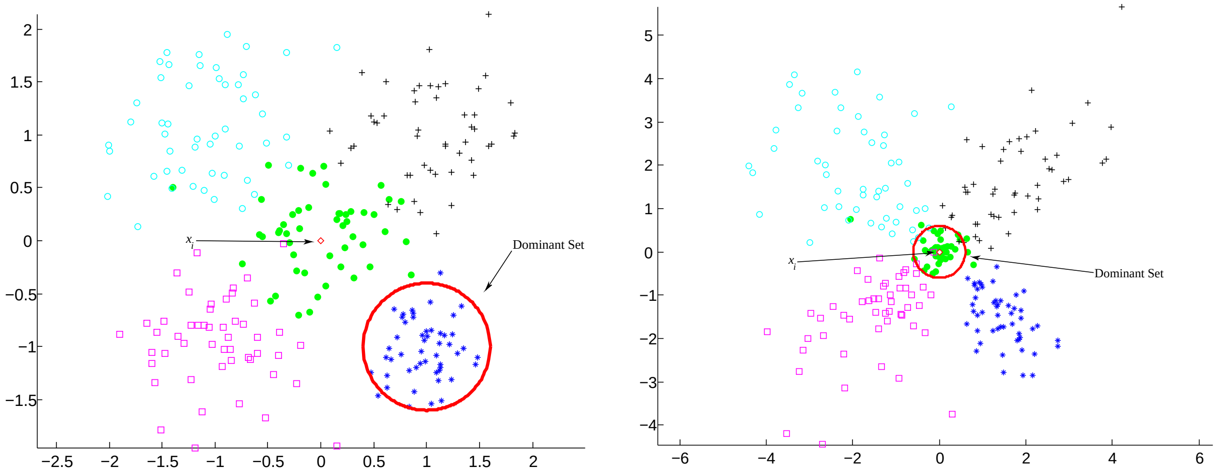
Unlike the pair-wise clustering in [21], there is a natural center point—x_i itself, associated with kNN(x_i) in a retrieval task. Starting from x_i, a closer point in kNN(x_i) is more likely to be a true neighbor of x_i. Therefore, we introduce a weight vector $\mathbf{u}_i = (u_{i1}, ..., u_{in})$ for kNN(x_i). The weight is defined as:

$$u_{ij} = \begin{cases} w_{ij}/T_i, & x_j \in k\text{NN}(x_i) \text{ or } j = i; \\ 0, & \text{otherwise;} \end{cases} \tag{7}$$

where $T_i = \sum_{x_j \in k\text{NN}(x_i) \text{ or } j=i} w_{ij}$ is a normalization factor. As demonstrated in Fig. 2, the dominant set of (a) is apart from x_i to the bottom-right dense area. In (b), the data points are weighted with their distances to x_i, where the dominant set shifts to the denser area surrounding x_i. Taking x_i itself into account, the objective is redefined as:

$$\begin{aligned} \max_{\mathbf{z}_i} \quad & f(\mathbf{z}_i) = \mathbf{z}_i^T (\sqrt{\mathbf{u}_i^T} W_0 \sqrt{\mathbf{u}_i}) \mathbf{z}_i = \mathbf{z}_i^T W_i' \mathbf{z}_i \\ & z_{ij} = 0, x_j \notin k\text{NN}(x_i) \cup \{x_i\}; \\ \text{s.t.} \quad & \sum_{j=1}^{n} z_{ij} = 1; \\ & z_{ij} \geq 0, j = 1, ..., n, \end{aligned} \tag{8}$$

where we have defined $W_i' = \sqrt{\mathbf{u}_i^T} W_0 \sqrt{\mathbf{u}_i}$. W_0 is transformed from W by setting the diagonal entries to zeros. Because if W is used directly, the solution can probably be



(a)

(b)

Fig. 2. Dominant sets in an non-weighted and weighted space.

$\mathbf{e}_i$ (the ith column of identity matrix I_n) as the similarity of i to itself is usually much larger than others.

The local maximizers of (8) can be obtained by the iterative method of [21]. At each iteration, the indicator vector $\mathbf{z}_i$ is updated by the following recursion equation:

$$z_{ij}(t+1) = z_{ij}(t) \cdot \frac{(W_i' \mathbf{z}_i(t))_j}{\mathbf{z}_i^T(t) W_i' \mathbf{z}_i(t)}, \; j = 1, ..., n. \quad (9)$$

Initialized with $\mathbf{z}_i = \mathbf{u}_i$, the objective function (8) is strictly increasing along the trajectory of (9) and converges to a local solution [21]. After (9) converges to a local solution $\mathbf{z}_i^*$, we obtain the dominant set $\mathrm{DS}(x_i)$ of the weighted $k\mathrm{NN}(x_i)$, where $x_j \in \mathrm{DS}(x_i)$ if and only if $z_{ij}^* > 0$. No matter how large the initial kNN is, the near neighbors of x_i with large weight are more likely to be selected by $\mathrm{DS}(x_i)$ than distant points with small weights. Let $x_{c(i)}$ and $x_{f(i)}$ be the closest and the farthest points in $\mathrm{DS}(x_i)/\{x_i\}$, respectively. To preserve patterns other than $\mathrm{DS}(x_i)$ in the class of x_i, we combine $\mathrm{DS}(x_i)$ with the ϵ-NN method. $\mathrm{DS}(x_i)$ is extended to $\mathrm{DS}'(x_i) = \{x_j | w_{ij} \geq \epsilon_i\}$, where the radius $\epsilon_i = w_{if(i)}$. Finally, $\mathrm{SAN}(x_i)$ is defined as:

$$\mathrm{SAN}(x_i) = \begin{cases} \mathrm{DS}'(x_i), & x_i \in \mathrm{DS}(x_i) \,\&\, \mathrm{Rank}(x_{c(i)}) \leq t_{1\mathrm{NN}}; \\ \bar{k}\mathrm{NN}(x_i), & \text{otherwise}; \end{cases} \quad (10)$$

where $\mathrm{Rank}(x_{c(i)}) \leq t_{1\mathrm{NN}}$ means the original rank of $x_{c(i)}$ is higher than the given threshold $t_{1\mathrm{NN}}$. We introduce the constraints in (10) to discard possible wrong neighborhoods. It is based on the intuition that if x_i itself is not included in $\mathrm{DS}(x_i)$ or $x_{c(i)}$ is far from x_i in the measure S, $\mathrm{DS}(x_i)$ may possibly depart from x_i to another denser class. We replace $\mathrm{DS}(x_i)$ with $\bar{k}\mathrm{NN}(x_i)$ in these cases, where $\bar{k}\mathrm{NN}(x_i)$ is the classic kNN of x_i, whose size $\bar{k} = \mathrm{mean}(\{\#\mathrm{SAN}(x_i) | x_i \in \mathrm{DS}(x_i), \mathrm{Rank}(x_{c(i)}) \leq t_{1\mathrm{NN}}\})$ is the mean size of the confident SANs.

Briefly, SAN is a variation of ϵ-NN, whose bound is not predefined but adapted by applying the dominant set in the weighted initial kNN. Thus, there is no need to specify the size of neighborhood for each dataset if we substitute SAN for kNN in the previous section.

V. The Affinity Matrix

In shape/image retrieval tasks, a distance metric often needs to be defined. The provided pairwise distance matrices need to be converted to affinity matrices before propagation. Moreover, the scale differences between distance matrices should be taken into account as they may cause problems to (3). Let $D = [d_{ij}]_{n \times n}$ be a distance matrix provided by some distance function. Usually it can be converted by applying a Gaussian kernel:

$$w_{ij} = \exp(-\frac{d_{ij}^2}{\sigma_{ij}^2}). \quad (11)$$

The scale of the kernel σ_{ij} can be selected by studying the local statistics of the neighborhood of x_i and x_j. A widely-used approach [22] adapts σ_{ij} based on the mean distance between x_i, x_j and their kNNs:

$$\sigma_{ij} = \alpha \cdot \mathrm{mean}(\{\mathrm{knnd}(x_i), \mathrm{knnd}(x_j)\}) \quad (12)$$

where $\mathrm{mean}(\{\mathrm{knnd}(x_i), \mathrm{knnd}(x_j)\})$ represents the mean distance of the k nearest neighbor distances of x_i and x_j, and α is an adjusting parameter. Both k and α are highly dependent on the dataset been used and are determined empirically.

By substituting SAN for kNN, we can adapt $k = \bar{k}$ for different datasets. The problem here is that the implementation of SAN is based on an affinity matrix and what we have is a distance matrix. We simply use $[d_{\max} - d_{ij}]_{n \times n}$ as the affinity matrix in this case, where $d_{\max}$ is the maximal element of D. Another parameter α is set to $\frac{1}{3}$ according to the three-sigma rule, which states that the mean distance is typically within $3\sigma_{ij}$.

VI. Experiments

We evaluate the proposed algorithm on shape/image retrieval and cluster analysis tasks. For both of them, we firstly

TABLE I
BULL'S EYE SCORES AND RETRIEVAL RATES OF TOP 20 RANKINGS ON MPEG-7 SHAPE DATABASE [16].

Algorithm	Bull's eye score	Retrieval rate of top 20 rankings
SC [2]	86.21%	79.20%
IDSC [1]	85.52%	77.12%
AIR [5]	93.67%	88.17%
LCDP [8] (IDSC)	92.36%	86.69%
LCDP+unsupervised-GP (IDSC)	93.32%	
LCDP+supervised-GP (IDSC)	97.21%	
Co-transduction [12] (SC+IDSC)	97.72%	95.62%
DN+TPG [6] (AIR)	99.99%	94.28%
LCMD+SAN (SC+IDSC)	99.67%	97.91%
LCMD+kNN (SC+IDSC)	98.84%	98.94%
LCMD+SAN (SC+AIR)	100%	99.44%
LCMD+kNN (SC+AIR)	100%	99.70%
LCMD+SAN (SC+IDSC+AIR)	100%	99.89%
LCMD+kNN (SC+IDSC+AIR)	100%	99.96%

TABLE II
1NN CLASSIFICATION ACCURACIES AND RETRIEVAL RATES OF TOP 150 RANKINGS ON SWEDISH LEAF DATASET [23].

Algorithm	1NN accuracy	Retrieval rate of top 150 rankings
SC [2]	94%	86.5%
IDSC [1]	94.13%	86.2%
LCDP [8] (IDSC)	98.2%	94.8%
LCDP+unsupervised-GP (IDSC)	97.6%	
LCDP+supervised-GP (IDSC)	99.3%	
Co-transduction [12] (SC+IDSC)		92.87%
TPG+DN [6] (IDSC)	97.33%	93.96%
LCMD+SAN (SC+IDSC)	97.87%	96.37%
LCMD+kNN (SC+IDSC)	98.4%	97.28%

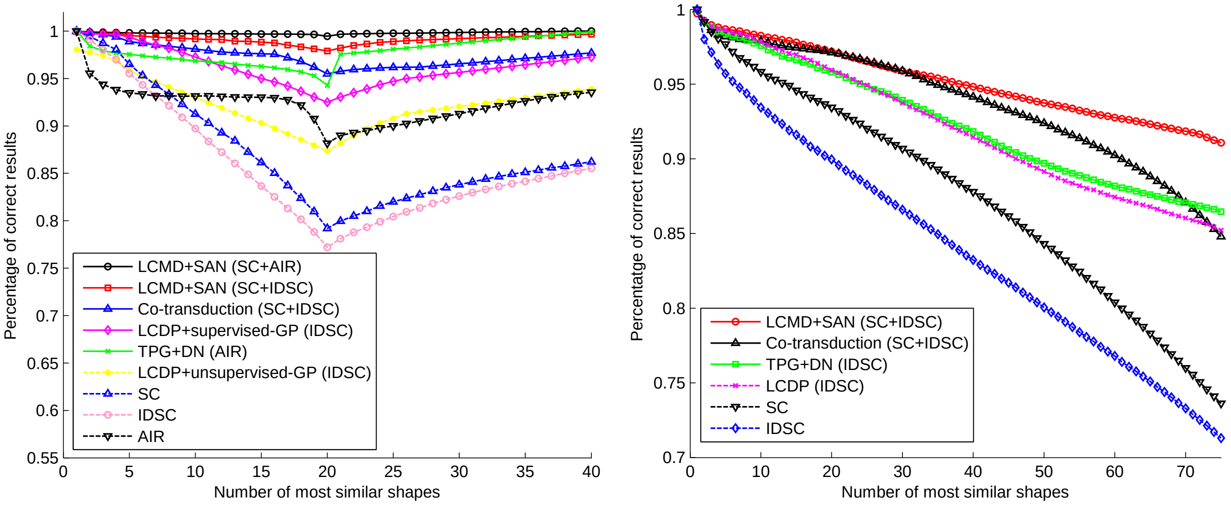
construct a single affinity matrix from two or three predefined measures with equal weights. Of course our LCDM framework does not limit to three input measures. In theory, the more measures we input and the less they correlate, the better LCMD may perform. In our experiments, the size k of the initial kNN and the threshold $t_{1\mathrm{NN}}$ for SAN is set to 100 and 5, respectively. The number of iterations t in the diffusion process is set to 15. Note that, if the input measures are highly correlated, the iterative convergence rate could be slow (discussed later in this section).

A. Shape and image retrieval

1) MEPG-7 shape database: The popular MPEG-7 CE-Shape-1 part B database contains 1400 silhouettes from 70 classes, where each class has 20 different shapes [16]. The retrieval accuracy is usually measured by the "bull's eye test", where every shape in the database serves as a query and the number of shapes from the same class among the top 40 rankings is counted. The bull's eye score is the ratio of all counted hits and the maximum possible number of correct hits (which is 20×1400). In the computation of SAN, the proportions of "otherwise" definition in (10) occurred are 5.6%, 4.5% and 0% for SC, IDSC and articulation-invariant representation (AIR) [5], respectively. Using SC and AIR as the original distance measures, the proposed LCMD+SAN method obtains a bull's eye score of 100%. To our knowledge, this is the best result that has been reported on this widely-used shape dataset. Table I summarizes several recent methods on the MPEG-7 database, where inputs of learning based methods are stated in brackets. The reported results of LCMD+kNN are obtained with the best manually tuned kNN. We see that SAN can produce comparable results without trial-and-error for the neighborhood size, and LCMD with three inputs performs better than two inputs. We also plot the percentage of correct results among the first k most similar shapes in Fig. 3(a) to visualize the gain in retrieval rates by our method. Since each class contains 20 shapes and the bull's eye test retrieves 40 shapes, the curves increase for $k > 20$. Thus, a bull's eye score of 100% does not mean perfect retrieval accuracy. Nevertheless, it is clear that the proposed method outperforms others in each ranking. TPG+DN obtains the second highest score on the bull's eye test, which is 0.01% lower. However, as shown in Table I, the top 20 rankings' retrieval rate of it (94.28%) is quite lower than the proposed method. The LCMD method performs much better in the stricter criterion, which demonstrates the effectiveness of the proposed method on high rankings.

2) Swedish leaf dataset: The Swedish leaf dataset comes from a leaf classification project at Linköping University and the Swedish Museum of Natural History [23], which contains 1125 leaf images from 15 different Swedish tree species, with 75 leaves per species. We ignore the appearance and only utilize the outline (shape) of a leaf. The 1-nearest-neighbor (1NN) classification are usually reported in previous work [1], [8], [23], where 25 leaves are used as training samples and others for testing per species. As shown in Table II, the best 1NN classification accuracy is obtained by adding supervised-GP in [8], while the best one of unsupervised methods is LCMD with SC and IDSC. We set the sizes of kNN to 18 for SC and IDSC to obtain the best accuracy. Substituted kNN with SAN, the accuracy is slightly lower than LCDP with IDSC. That is because SAN, whose greatest advantage is its self-adaptation, is an acceptable but not the best definition of neighborhood. In the computation of SAN, the "otherwise" definition in (10) account for 9.2% and 7.1% to all data points for SC and IDSC, respectively. We also report the retrieval rates of the top 150 (75×2) rankings similar to the bull's eye test in the table, where LCMD with manually specified kNN also takes the highest score (97.28%) and LCMD+SAN is in the second place (96.37%). Fig. 3(b) shows the retrieval curves of the top 75 rankings. It is clear that our proposed approach consistently retrieves most relevant leaves in each ranking compared with other methods, because LCMD could refine the order of rankings at the same time of retrieving more.

3) Nister and Stewenius dataset: We then evaluate the proposed approach on the Nister and Stewenius (N-S) image dataset [24]. N-S dataset consists of 2,550 objects or scenes, each of which has 4 images in different viewpoints. The task is to find the copies of each query in the totally 10,200 images. The result is evaluated by the average number of corrects in the top 4 returning images. Thus the highest possible score is 4. Due to the sparsity of data space, diffusion based manifold learning approaches are difficult to implement in this dataset.



(a)

(b)

Fig. 3. The curves of retrieval rates on MPEG-7 shape dataset [16] (a) and Swedish Leaf dataset [23] (b).

TABLE III
RETRIEVAL RESULTS ON N-S DATASET [24].

Algorithm	Base			N-S score
CMD [17]	3.26 (1 visual vocabulary) 3.33 (19 visual vocabularies)			3.57 3.68
Co-transduction [12]	3.26			3.66
TPG+DN [6]	3.22			3.61
LCMD+SAN	3.20[1]	3.17[2]	2.81[3]	3.61 ($t = 15$) 3.65 ($t = 100$)
LCMD+kNN				3.70

[1] http://bigimbaz.inrialpes.fr/herve/share/ukb_k30000_bof.gz;
[2] http://vis.uky.edu/∼stewe/ukbench/data/vw_ukbench610_normal.zip;
[3] http://vis.uky.edu/∼stewe/ukbench/database_as_visual_words.zip.

Even such, recent studies [6], [12], [17] make significantly improvement as shown in Table III. The bases of the proposed algorithm are three vocabulary tree methods. One is introduced in [17] with a 30,000 size vocabulary, and the left two in [24] with different 1,000,000 size vocabularies. The percentages of "otherwise" definition of (10) in the computation of SAN are 5.7%, 1.2% and 5.0% for the three inputs, respectively. As shown in the table, LCMD achieves the highest score (3.70) by fixing the neighborhood sizes of inputs to 7. CMD [17] obtains the second highest score (3.68) while 19 distinct visual vocabularies are used in it in contrast to 3 in the proposed algorithm. The similar computation processes of the original measures make the input affinity matrices highly correlated, which slows down the iterative convergence rate of LCMD. We have set $t = 100$, which gains a better score (3.65) in LCMD+SAN than the predefined $t = 15$ (3.61). The score is a little lower than LCMD with manually tuned kNN. It may be due to the fact that the compact cluster searched by (8) is somewhat random when the size of each class in this dataset (4 images per class) is so small. However, the result is still comparable with other methods when it is no need to trial-and-error for the neighborhood size.

4) Caltech-101 dataset: The number of instances per class is constant in the previous datasets. We also test our method on the well-known diverse Caltech-101 dataset [25], which contains 101 classes with 8,677 images in total. The size of each class varies from 31 to 800. Adapted from [6], we use a subset consisted of 2,788 images from 12 classes as examples shown in Fig. 4. Each image is represented by multiple assignments [17] and the spatial pyramid method [26] based on a 2048 size SIFT [27] codebook. The distance between two images is defined as the χ^2 distance of their representation vectors. Since LCMD needs at least two measures, we generate the reverse distance (adapted from the reverse similarity of [12]) of the previous one. Let D_{ij} denote the original χ^2 distance of image i and j, the reverse distance D^r_{ij} is the ranking number of i when using j as a query. The reverse distance represents the symmetrical character of the original measure. When two images are both in the top rankings using the other one as the query, they are more likely to be from the same category. Since the numerical value of the reverse distance is only a reflection of the relation between images, we use the corresponded SAN of the original distance in the determination of the reverse distance's neighborhood size. In the computation of SAN the proportion of "otherwise" definition in (10) occurred is 10.3%. The results are shown in Table IV, where the retrieval accuracy is defined as the mean ratio of correct hits in the top K (K is the size of each query's class) retrieval results. Obviously, LCMD outperforms other methods in the retrieval rate. Since the reverse distance is derived from the χ^2 distance, the iterative convergence rate slows down on this dataset. However, the retrieval accuracy of LCMD+SAN with $t = 15$ (94.86%) is already better than TPG+DN [6] (91.53%), while the result with $t = 100$ (98.23%) is comparable to LCMD with manually tuned kNN (98.67% with $k = 36$).

B. Cluster analysis

Besides shape retrieval, the learned affinity matrix by the proposed approach can also be used for cluster analysis. We apply Affinity Propagation (AP) [28] on the learned affinity matrixes of MPEG-7, Swedish Leaf and Caltech-101 datasets,

Fig. 4. Examples (two images for each class) from the selected subset of the Caltech-101 dataset [25].

TABLE IV
Retrieval rates on 12 image classes from the Caltech-101 dataset [25].

Base	TPG+DN	LCMD+kNN	LCMD+SAN ($t = 15$)	LCMD+SAN ($t = 100$)
83.82%	91.53%	98.67%	94.86%	98.23%

TABLE V
Clustering performance on MPEG-7 shape database [16]

Algorithm	TPG+DN (AIR)	Co-trained spectral [29] (SC+AIR)	LCMD+SAN (SC+AIR)	LCMD+kNN (SC+AIR)
F-score	0.937	0.900 (0.019)	0.993	0.999
NMI	0.971	0.981 (0.004)	0.997	0.999

TABLE VI
Clustering performance on Swedish Leaf dataset [23]

Algorithm	TPG+DN (IDSC)	Co-trained spectral (SC+IDSC)	LCMD+SAN (SC+IDSC)	LCMD+kNN (SC+IDSC)
F-score	0.828	0.801 (0.045)	0.834	0.830
NMI	0.912	0.892 (0.015)	0.929	0.933

TABLE VII
Clustering performance on 12 image classes from the Caltech-101 dataset [25]

Algorithm	Base	TPG+DN [6]	LCMD+SAN ($t = 15$)	LCMD+SAN ($t = 100$)	LCMD +kNN
F-score	0.564	0.640	0.777	0.745	0.745
NMI	0.713	0.769	0.885	0.885	0.885

respectively. For the convenience of comparison, the number of clusters in all experiments are set to the ground truth. Two evaluation measures—normalized mutual information (NMI) and F-score, are reported. For both of them, the higher value indicates better clustering quality in some sense.

1) MEPG-7 shape database: Two references, TPG for diffusion methods and co-trained spectral clustering [29] for multi-view methods are compared with the proposed algorithm in this section. TPG and LCMD are both learning based methods, which produce a new affinity matrix as output. The original distance measure of TPG is AIR, and the measures of LCMD are AIR and SC. We apply AP on the learnt affinity matrixes for clustering. In contrast, co-trained spectral clustering, whose output is cluster labels, is specially designed for multi-view clustering and does not need AP anymore. Table V shows the clustering performance of the three methods. Since co-trained spectral clustering employs k-means in the last step, the performance measures of which are the means and standard deviations (in the brackets) with 20 different runs of k-means with random initializations as in [29]. It is clear that the proposed LCMD approach, with SAN or manually set kNN, outperforms the other two with very high scores in both F-score and NMI. Actually, only 5 and 1 out of 1400 shapes are mis-clustered in LCMD+SAN and LCMD+kNN, respectively. It is interesting that the single view method (TPG) performs better than the multi-views method (co-trained spectral clustering) in F-score. It may be because that co-trained spectral clustering throws away the within cluster details of the origin distance matrix, which may drop the effectiveness of clustering.

2) Swedish Leaf Dataset: Clustering on Swedish Leaf dataset is harder than the previous task because the items in this dataset are all less discriminative leaves. By substituting IDSC with AIR, the experiment setting on this dataset is the same as that on MPEG-7. Table VI summarizes the experiment results of the three methods. Again, our LCMD approach outperforms the competitors.

3) Caltech-101 dataset: Multiple measures are both utilized in the previous two experiments. We also test the proposed algorithm with a single measure on the subset of Caltech-101 dataset. All settings are the same as in the retrieval experiment. We report the clustering performances of the base χ^2 distance measure, TPG learning, and our LCMD method in Table VII. It can be seen that both TPG and LCMD can improve the χ^2 distance measure, while our method is better. As mentioned in the retrieval experiment, the iterative convergence rate slows down due to the use of "reverse distance" in this dataset. When in the clustering experiments, a small number of t is sufficient as shown in the table.

VII. Conclusion

In this paper, we present a locally constrained mixed-diffusion method for shape/image retrieval. We not only utilize the union of different measures to recall more correct results, but also make use of the intersection of them to refine the order of top retrieval results. We have also provided a new definition of neighborhood whose size is self-tuned. Experiments on both shape and image datasets demonstrate the effectiveness of the proposed approach, and also show the potential use in other computer vision tasks, *e.g.* cluster analysis. Recently, Liu *et al.* [30] proposed k-dense neighborhood, which is a new dense subgraph detection method. The advantage of k-dense neighborhood is that it can control the size of the dense subgraph. We are going to use it to substitute the dominant set in the construction of SAN. Since our approach is generic and not limited to shape/image retrieval, we will extend it to a broad range of retrieval and matching problems in computer vision in the future.

A limitation of this work is its high time complexity. In the construction of SAN, the main computation load is the replicator dynamics procedure of (9). Suppose the average number of iterations for the equation is t_{DN}, its time complexity is $O(kt_{\mathrm{DN}})$, where $k = 100$ is the size of initial kNN. The diffusion process in LCMD needs to multiply the simiarity matrices as in (5), which is $O(n^3)$. Thus, to query each object in the dataset once, the total time complexity of LCMD+SAN, similar to [6] and [8], is $O(n^3t + kt_{\mathrm{DN}})$ for the whole dataset, or $O(n^2t + \frac{kt_{\mathrm{DN}}}{n})$ for each query on average, where $t = 15$ is the iteration times of (5). In a practical retrieval system, there is no need to compute the affinity matrix of the entire dataset, and we can construct input matrices only using the first $N \ll n$ most similar objects for each measure as in [10] and [12]. Then the matrix size n is less than mN, where m is the number of input measures.

Acknowledgment

The participation of L. Luo and C. Zhang was in part supported by NSFC projects 61033008 and 60903041. The participation of C. Shen and A. van den Hengel was in part support by ARC grant LP120200485. C. Shen was also supported by ARC Future Fellowship FT120100969.

References

[1] H. Ling and D. W. Jacobs, "Shape classification using the inner-distance," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 29, no. 2, pp. 286–299, 2007.

[2] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 24, no. 4, pp. 509–522, 2002.

[3] P. Felzenszwalb and J. Schwartz, "Hierarchical matching of deformable shapes," in *Proc. IEEE Conf. Comp. Vision. Pattern Recogn.*, 2007.

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

[5] R. Gopalan, P. Turaga, and R. Chellappa, "Articulation-invariant representation of non-planar shapes," in *Proc. Eur. Conf. Comp. Vision*, 2010.

[6] X. Yang and L. J. Latecki, "Affinity learning on a tensor product graph with applications to shape and image retrieval," in *Proc. IEEE Conf. Comp. Vision. Pattern Recogn.*, 2011.

[7] D. C. G. Pedronette and R. D. S. Torres, "Exploiting contextual spaces for image re-ranking and rank aggregation," in *Proc. ACM Int. Conf. Multimedia Retrieval*, 2011.

[8] X. Yang, S. Koknar-Tezel, and L. J. Latecki, "Locally constrained diffusion process on locally densified distance spaces with applications to shape retrieval," in *Proc. IEEE Conf. Comp. Vision. Pattern Recogn.*, 2009, pp. 357–364.

[9] P. Kontschieder, M. Donoser, and H. Bischof, "Beyond pairwise shape similarity analysis," in *Proc. Asian Conf. Comp. Vision*, 2009, pp. 655–666.

[10] X. Yang, X. Bai, L. J. Latecki, and Z. Tu, "Improving shape retrieval by learning graph transduction," in *Proc. Eur. Conf. Comp. Vision*, 2008.

[11] J. Huang, X. Yang, X. Fang, W. Lin, and R. Zhang, "Integrating visual saliency and consistency for re-ranking image search results," *IEEE Trans. Multimedia*, vol. 13, no. 4, pp. 653–661, 2011.

[12] X. Bai, B. Wang, X. Wang, W. Liu, and Z. Tu, "Co-transduction for shape retrieval," in *Proc. Eur. Conf. Comp. Vision*, 2010.

[13] D. Zhou and C. Burges, "Spectral clustering and transductive learning with multiple views," in *Proc. Int. Conf. Mach. Learn.*, 2007, pp. 1159–1166.

[14] S. Santini and R. Jain, "Similarity measures," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 21, pp. 871–883, 1999.

[15] V. Mezaris, S. Gidaros, W. Kasper, J. Steffen, R. Ordelman, M. Huijbregts, F. de Jong, I. Kompatsiaris, and M. Strintzis, "A system for the semantic multimodal analysis of news audio-visual content," *EURASIP Journal on Advances in Signal Processing*, vol. 2010, p. 47, 2010.

[16] L. J. Latecki, R. Lakaemper, and U. Eckhardt, "Shape descriptors for non-rigid shapes with a single closed contour," in *Proc. IEEE Conf. Comp. Vision. Pattern Recogn.*, 2000, pp. 424–429.

[17] H. Jegou, C. Schmid, H. Harzallah, and J. Verbeek, "Accurate image search using the contextual dissimilarity measure," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 32, no. 1, pp. 2–11, 2010.

[18] S. Santini, A. Gupta, and R. Jain, "Emergent semantics through interaction in image databases," *IEEE Trans. Knowledge & Data Engineering*, vol. 13, no. 3, pp. 337–351, 2001.

[19] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," in *Proc. Conf. Learning Theory*, 1998, pp. 92–100.

[20] R. R. Coifman and S. Lafon, "Diffusion maps," *Applied and Computational Harmonic Analysis*, vol. 21, no. 1, pp. 5–30, 2006.

[21] M. Pavan and M. Pelillo, "Dominant sets and pairwise clustering," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 29, no. 1, pp. 167–172, 2007.

[22] J. Wang, S.-F. Chang, X. Zhou, and S. T. C. Wong, "Active microscopic cellular image annotation by superposable graph transduction with imbalanced labels," in *Proc. IEEE Conf. Comp. Vision. Pattern Recogn.*, 2008.

[23] O. Soderkvist, "Computer vision classification of leaves from swedish trees," Master's thesis, Linköping University, Sweden, 2001.

[24] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," in *Proc. IEEE Conf. Comp. Vision. Pattern Recogn.*, 2006.

[25] L. Fei-Fei, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories," in *IEEE Conf. Comp. Vis. Patt. Recogn., Workshop Generative-Model Based Vision*, 2004.

[26] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Proc. IEEE Conf. Comp. Vision. Pattern Recogn.*, 2006.

[27] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comp. Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[28] B. Frey and D. Dueck, "Clustering by passing messages between data points," *Science*, vol. 315, no. 5814, pp. 972–976, 2007.

[29] A. Kumar and H. Daumé III, "A co-training approach for multi-view spectral clustering," in *Proc. Int. Conf. Mach. Learn.*, 2011.

[30] H. Liu, X. Yang, L. Latecki, and S. Yan, "Dense neighborhoods on affinity graph," *Int. J. Comp. Vision*, 2012.