

Shape and Image Retrieval by Organizing Instances Using Population Cues

Andrew Temlyakov, Pahal Dalal, Jarrell Waggoner, Dhaval Salvi, and Song Wang
Department of Computer Science and Engineering
University of South Carolina, Columbia, SC 29208, USA

{temlyaka, dalalpk, waggonej, salvi}@email.sc.edu, songwang@cec.sc.edu

Abstract

Reliably measuring the similarity of two shapes or images (instances) is an important problem for various computer vision applications such as classification, recognition, and retrieval. While pairwise measures take advantage of the geometric differences between two instances to quantify their similarity, recent advances use relationships among the population of instances when quantifying pairwise measures. In this paper, we propose a novel method which refines pairwise similarity measures using population cues by examining the most similar instances shared by the compared shapes or images. We then use this refined measure to organize instances into disjoint components that consist of similar instances. Connectivity is then established between components to avoid hard constraints on what instances can be retrieved, improving retrieval performance. To evaluate the proposed method we conduct experiments on the well-known MPEG-7 and Swedish Leaf shape datasets as well as the Nister and Stewenius image dataset. We show that the proposed method is versatile, performing very well on its own or in concert with existing methods.

1. Introduction

Identifying the relationships among a set of shapes or images (instances) is a fundamental problem in computer vision. This problem is related to many tasks such as retrieval, classification, and recognition which are applied to various real-world problems. For example, the species of trees or plants can be determined by identifying and classifying the shape of their leaves [4]. However, identifying the relationships among instances, be they images or shapes, is a very challenging problem. While humans can easily describe the shape of an object subjectively, it is very difficult to develop measures that can objectively and consistently describe the shape or appearance of a given object. Furthermore, the problem becomes more challenging due to the large variation observed in shapes or images that describe the same class and the presence of noise and outliers. For example,

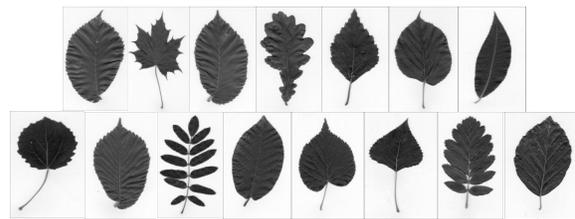


Figure 1. Example of 15 leaf instances, each representing a unique class in the Swedish Leaf dataset.

automatically classifying the Swedish Leaf dataset [22] into the 15 ground-truth classes, which are shown in Fig. 1, is a well-known, challenging problem.

Many previous works have proposed shape-based measures and algorithms to describe and compare a given population of shapes [1, 6, 24, 19, 8]. More recently [13, 12, 15, 14] introduce new methods which find the similarity between two shapes, as defined by various shape-based measures, with greater accuracy. Specifically, Ling and Jacobs [13] introduce Inner-Distance Shape Context (IDSC) which further improves the widely-known Shape Context descriptor [5], while [14] introduces a framework called Aspect Space which aims to balance deformability and discriminability for describing a shape. Further, Liu *et al.* [15] apply Total Bregman Divergence to characterize the similarity between two shapes and classify them into groups of similar shapes, while Lin *et al.* [12] introduce the Layered Graph Matching algorithm. In addition to these measures, Temlyakov *et al.* [23] present the “strand” and “stretch” strategies (S&S) that simplify the geometry of a given shape to improve the performance of these shape descriptors and similarity measures. These previous works have made significant advancements in quantifying the similarity between a given pair of shapes. However, the major drawback of these descriptors is their susceptibility to large variations and outliers because of the myopic nature of comparing features from only the two instances under consideration.

Recently, [25, 3, 10, 2, 26] suggest methods that exploit the implicit relationships among a collection of shapes based on the previously developed methods. In this way,

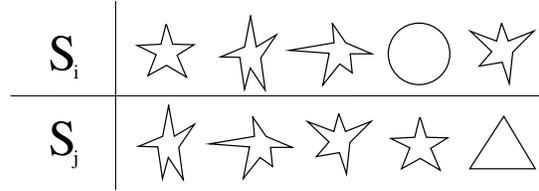
they better describe the similarity between a pair of shapes by including information that goes beyond the pairwise formulation. Specifically, Yang *et al.* [25] introduce the Locally Constrained Diffusion Process (LCDP) over constrained neighborhoods of graph nodes, each representing a shape, to better describe the similarity between a pair of instances. Further, they introduce “ghost points” to densify the dataset and better represent the underlying shape manifold structure. Similarly, Kotschieder *et al.* [10] construct a Modified Mutual k -Nearest Neighbors (MM- k NN) graph using one of the previously developed pairwise shape similarity measures to represent the underlying relationship among shapes in a collection. This graph is used to identify better matches for a given query shape for shape retrieval. In [3], Bai *et al.* present a Label Propagation (LP) method, where the similarity measure is influenced by the k -closest matching instances of a query shape. Yang *et al.* [26] further improve propagation methods by obtaining a product tensor graph from the tensor product of the original graph with itself. The similarity information is then propagated along the product tensor graph instead of the original graph. Finally, [2, 20] propose frameworks which combine the similarity measures obtained from previous methods (*e.g.*, from [5, 13, 14, 19]).

Image-based retrieval employs such techniques as bag-of-features [21, 18] to quantify the similarity of two images (instances). These pairwise methods suffer from the same downsides as the shape retrieval methods, because they only consider information in the two images alone, and not the neighborhood of other images that are mutually similar to the two images being considered. Jegou *et al.* [9] implicitly incorporate population information by ensuring that the query and a retrieved image are mutually similar. However, the method does not directly compare other images to explicitly incorporate global cues.

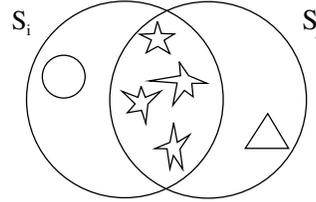
In this paper, we propose a novel method which: (1) extracts population cues by analyzing the relationships between any two instances under consideration, together with the remaining instances in the population, explicitly incorporating global information about the population, and (2) uses this refined information to organize similar instances into components. To do this, we compare two instances by using their respective closest matches, reducing the problem to one of set comparison. Shared closest matches provide a basis for organizing instances into components. We can then derive a final similarity for any two instances that is based on both the individual similarity between the instances and the similarity of their respective components.

2. Population Cues

Consider a set of N instances $S = (S_1, S_2, \dots, S_N)$. The aim of a similarity measure $d(S_i, S_j)$ is to reliably define the similarity between the two instances S_i and S_j . In



(a) The closest 5 matches of two instances, S_i and S_j .



(b) The overlap of above two sets indicates how similar S_i and S_j are to each other.

Figure 2. Motivation for utilizing population cues: truly similar instances must have significant overlap in k -closest matches.

previous works, various measures such as [13, 12, 24, 15, 14] have been proposed for shapes. However, most of these only consider the similarity in shape-based features of a pair of shape instances while calculating $d(S_i, S_j)$ and increase the chances that two shape instances will incorrectly be characterized as very similar (or dissimilar). In [25, 3, 10] it is suggested that beginning with one of these previous measures $d(S_i, S_j)$, a more global analysis of each individual instance may lead to a better, updated measure $d_u(S_i, S_j)$. With some relaxation of notation, we refer to any such measure, whether $d(S_i, S_j)$ or $d_u(S_i, S_j)$, obtained from any of the above mentioned methods as $d(S_i, S_j)$ for the remainder of this paper.

We derive the similarity of two instances S_i and S_j through the overlap in their respective k -closest matches based on any arbitrary similarity measure $d(S_i, S_j)$. For example, consider the two instances S_i and S_j in Fig. 2(a). In the second column of each row, we illustrate the 5 closest matches to these instances as described by some similarity measure d . Since the sets of closest matches have a significant overlap, as illustrated in Fig. 2(b), we say that the two instances are very similar even if the value of $d(S_i, S_j)$ may suggest otherwise. In this way, we do not simply depend on measures based on extracted features to describe the similarity between two instances. Further, if two instances have high similarity based on a measure $d(S_i, S_j)$ but very low agreement in their closest matches, we can conclude that the high similarity through $d(S_i, S_j)$ is purely incidental.

More formally, we can represent the set of k -closest matches of S_i by B_i , and similarly the k -closest matches of S_j by B_j . To reliably quantify the similarity between two sets, we use the widely-known Dice coefficient. Let the intersection of the sets of k -closest matches, B_i and B_j ,

be represented by $B_i \cap B_j$. We formally define population agreement between two shape instances as:

$$\rho(S_i, S_j) = 1 - \frac{2|B_i \cap B_j|}{|B_i| + |B_j|}, \quad (1)$$

where $|\cdot|$ calculates the set size. Since the Dice coefficient always has a value between 0 and 1, so will ρ . In this way, $\rho(S_i, S_j) = 1$ for instances that are very *dissimilar* and $\rho(S_i, S_j) = 0$ if they show high level of similarity. In this way, we can encode broad population cues into simple pairwise measures.

2.1. Automatic Selection of k

In this paper we automatically choose a unique k for each pair of instances by considering the ordering of S_i and S_j based on d . We first order all instances in ascending order by their values of d with respect to a given instance S_i . The value of k is then determined by the position of S_j in this sorted list. In other words, k is set to be the number of instances that have higher similarity to S_i than S_j plus 1. To avoid problems caused by very large or small values of k , we introduce an upper bound b_u and a lower bound b_l . In this way, a suitable value for k is automatically selected in the range $[b_l, b_u]$ based on the pair of instances S_i and S_j rather than selecting a fixed value for all pairs.

3. Components of Similar Instances

Having calculated the value of $\rho(S_i, S_j)$ based on population cues for each pair of instances, S_i and S_j , we model the entire population as a graph to easily construct components which contain similar instances. The entire population is then considered as an undirected, fully-connected graph $G = (V, E)$ where $V = (v_1, v_2, \dots, v_N)$ represent each instance S_1, S_2, \dots, S_N and each edge $e_{ij} \in E$ connects the vertices v_i and v_j . We can assign the value $\rho(S_i, S_j)$ as the weight of each edge e_{ij} so that instances with a very strong relationship are connected to each other by edges with smaller edge weight, while instances that are very dissimilar are connected by edges with high edge weights. To construct meaningful components, we first eliminate all edges that connect instances that are very dissimilar. By selecting a suitable threshold t , we can easily eliminate any such edges e_{ij} .

The major problem then, is to select a suitable value for t . If the edges are weighted by the original similarity measure d , it may be difficult to identify an appropriate threshold t . However, using the measure ρ , developed in Section 2, it is simple to determine a suitable value for t . Specifically, we assume that any two instances that are similar enough to be included in a component must have high overlap in their respective k -closest matches and, hence, a

low value for ρ . In fact, $t = 0$ (100% overlap in the k -closest matches of S_i and S_j) may be a reasonable threshold, as it ensures that only the most similar instances are connected by an edge, and are thus able to be grouped together in a component. However, this value of t can lead to components which are very compact, and many components may only contain a single instance, which would prevent other, similar instances from influencing their similarity measure. Thus, we select $t \leq 0.3$ as our threshold and eliminate any edges with weight greater than t . In this way, we reduce the possibility of incorrectly grouping dissimilar instances in a component.

The components can then be calculated from the resulting graph G' using a single linkage algorithm, where two instances S_i and S_j are in the same component if, and only if, there exists a chain $S_i, S_{i1}, S_{i2}, \dots, S_{in}, S_j$ such that S_i has an edge with S_{i1} , S_{i1} has an edge with S_{i2} , and so on, for the rest of the chain [7].

The entire process of constructing components is illustrated in Fig. 3. A sample edge list, where $w(e_{ij}) = \rho(S_i, S_j)$, is shown in Fig. 3(a). Figure 3(b) shows the resulting fully connected graph, where each node represents a unique instance, and dashed lines indicate edges where $w(e_{ij}) > t$. Finally, Fig. 3(c) shows the resulting edges after pruning edges with weights higher than the threshold $t = 0.3$ and constructing the components, which are indicated by dotted lines.

4. Instance Similarity

Using the components constructed by grouping similar instances, we define the similarity between two instances S_i and S_j as

$$\psi(S_i, S_j) = r_c(S_i, S_j) + \rho(S_i, S_j), \quad (2)$$

where r_c depends on the components containing S_i and S_j .

Each connected component C , can be considered a single cohesive instance, since the instances belonging to the same component are very similar to each other. A pairwise similarity $\phi(C_p, C_q)$ can be found between any two components, C_p and C_q . To compute this similarity between components, we find the mean of all the pairwise values of $\rho(S_i, S_j)$ where $S_i \in C_p$ and $S_j \in C_q$. More specifically, we define $\phi(C_p, C_q)$ as

$$\phi(C_p, C_q) = \frac{1}{m \cdot n} \sum_{S_i \in C_p} \sum_{S_j \in C_q} \rho(S_i, S_j), \quad (3)$$

where m is the number of instances in C_p and n is the number of instances in C_q . In this way, each instance in a component has an equal influence on the other instances in the same component.

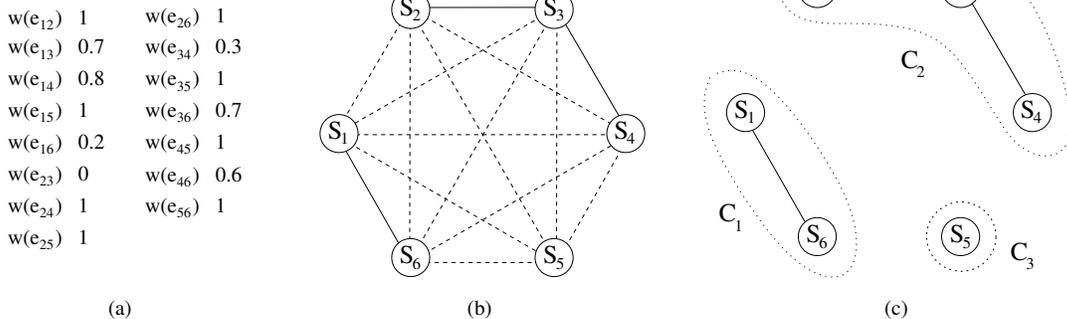


Figure 3. An illustration of constructing components. (a) An edge list for a graph of six nodes, where each node corresponds to a unique instance. (b) The resulting fully connected graph, where dashed edges represent edges with $w(e_{ij}) > t$, $t = 0.3$. (c) Graph after pruning all edges with $w(e_{ij}) > t$ and identifying the resulting components, shown by dotted lines.

We do not directly define $r_c(S_i, S_j)$ as ϕ . For an instance $S_i \in C_p$, we rank all components in increasing order of their dissimilarity to C_p and set $r_c(S_i, S_j)$ as the rank of component C_q , which contains S_j , in this ordering. If the two instances belong to the same component, we set $r_c(S_i, S_j) = 0$. The main reason is that we want r_c to be the more influential factor in the overall similarity measure ψ since it is a stronger measure than $\rho(S_i, S_j)$.

Sorting components according to r_c does not impose any ordering upon the instances within a component. By including $\rho(S_i, S_j)$ in our similarity measure, we can impose an ordering on instances that belong to the same component. It is then clear that the retrieval rate will depend entirely on $\rho(S_i, S_j)$ when the entire population of instances is treated as a single component. Similarly, when each instance belongs to its own unique component, $r_c(S_i, S_j)$ is also determined by $\rho(S_i, S_j)$ alone.

5. Experiments

Similar to most recent works related to retrieval and classification, we use the MPEG-7 dataset (specifically the MPEG-7 CE-Shape-1 Part B) [11], the Swedish Leaf dataset [22], and the Nister and Stewenius (N-S) image dataset [18] for our experiments. We compare our method with four recently developed methods: (1) Label Propagation (LP) [3], (2) Locally Constrained Diffusion Process (LCDP) [25], (3) Modified Mutual k -Nearest Neighbors (MM- k NN) [10], and (4) Tensor Product Graph with Dominant Neighbors (DN-TPG) [26]. For LP, LCDP, and DN-TPG we have maintained the original parameters as reported by their respective publications. For the MM- k NN method on the MPEG-7 dataset, we change this method’s parameters to $kPer = 13$, $nn = 10$, and $mn = 2$ after the authors [10] suggested that these parameters lead to better performance when used with S&S and IDSC (S&S+IDSC). As shown by the discussion in the following sections, we obtain comparable performance in shape or image retrieval

when compared to these state-of-the-art methods. Furthermore, in conjunction with any of these methods, our experiments reveal that the accuracy of retrieval can be improved by the proposed method.



Figure 4. Shape instances representing a subset of the 70 shape classes in the MPEG-7 dataset.

MPEG-7 Dataset: The MPEG-7 dataset consists of 1400 shape instances representing 70 ground-truth classes, each having 20 instances. A subset of the 70 shape classes are shown in Fig. 4. The *Bullseye* test is often used to evaluate the performance of shape retrieval on the MPEG-7 dataset. In this test, a shape instance is selected from the 1400 as the template and the 40 most similar instances to this template, as defined by a shape similarity measure, are retrieved. The number of retrieved instances belonging to the same class as the template is divided by 20 (the total number of shape instances in each class) to obtain the retrieval accuracy for this single template. In this way, the overall retrieval accuracy is calculated by selecting each of the 1400 instances as the template and finding the average for all of them.

For the proposed method, we set the upper bound for selecting k to $b_u = 13$ and the lower bound to $b_l = 4$ when finding the k -closest matches in our experiments. The threshold for finding components is set to $t = 0.3$ as explained previously. To illustrate performance improvement, we select the basic S&S+IDSC measure introduced in [23] as the base similarity measure, $d(S_i, S_j)$. Table 1 shows the performance of our framework compared with LP, LCDP, MM- k NN, and DN-TPG where our method obtains an accuracy of 96.36%, which is the best on the traditional Bullseye test when SS+IDSC is selected as $d(S_i, S_j)$. From Ta-

ble 1 we can also observe that a method which performs well on the top 40 measure may not be satisfactory for the strict top 20 measure, which is the ground-truth number of instances in each category in the MPEG-7 dataset. In addition, we focus on the best 20 matches in the modified Bullseye test to better delineate shape similarity evaluation measures, since the accuracy reported by the traditional test may not clearly indicate the advantages of one method over another. Using this modified Bullseye test, we find that the proposed method shows very good results, with only the DN-TPG method achieving better results.

Method	Top 40	Δ	Top 20	Δ
S&S + IDSC + Our Method	96.36 %	7.97 %	91.47 %	11.83 %
S&S + IDSC + DN-TPG	96.24 %	7.85 %	92.76 %	13.12 %
S&S + IDSC + MM- k NN	96.04 %	7.65 %	87.26 %	7.62 %
S&S + IDSC + LCDP	95.60 %	7.21 %	89.74 %	10.1 %
S&S + IDSC + LP	94.29 %	5.9 %	89.93 %	10.29 %
S&S + IDSC	88.39 %	0.0 %	79.64 %	0.0 %

Table 1. Performance of the proposed method, LP, LCDP, MM- k NN, and DN-TPG methods on the S&S+IDSC metric for the MPEG-7 shape dataset.

Method	Top 40	Δ	Top 20	Δ
DN-TPG + Our Method	95.67 %	-0.57 %	92.99 %	0.23 %
MM- k NN + Our Method	96.86 %	0.82 %	92.81 %	5.55 %
LCDP + Our Method	96.24 %	0.64 %	92.95 %	3.21 %
LP + Our Method	94.08 %	-0.21 %	91.29 %	1.36 %

Table 2. Performance of the proposed method when applied to results obtained by LP, LCDP, MM- k NN, and DN-TPG on the MPEG-7 dataset using the SS+IDSC as the base similarity measure.

As mentioned in Section 2, one of the strengths of our method is that it can be used in concert with the LCDP, LP, MM- k NN, or DN-TPG methods by using them as $d(S_i, S_j)$. As shown in Table 2, our method together with any of these methods can significantly improve the retrieval accuracy on the modified Bullseye test when retrieving the top 20 shape instances. Using the proposed method along with MM- k NN, we can improve the retrieval accuracy by over 5%.

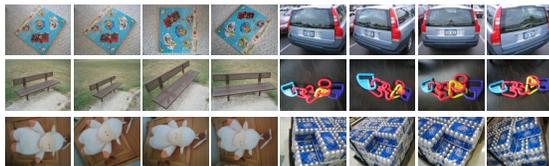


Figure 5. Image instances representing a subset of the 2, 550 classes in the N-S dataset.

Swedish Leaf Dataset: The Swedish Leaf dataset contains leaves from 15 different species of trees with 75 leaves per species, for a total of 1125 leaf-shape instances. The 15 leaf species are illustrated in Fig. 1. For all our experiments

on this dataset, we set $b_u = 47$, $b_l = 11$ and once again set $t = 0.3$. However, for this dataset, we directly use the IDSC measure as $d(S_i, S_j)$ since the S&S strategy is shown to not have any significant impact on the retrieval accuracy for this dataset [23]. Because we are using the original IDSC measure, we choose the default parameters as reported in [10] for MM- k NN. Specifically we set $kPer = 8$, $nn = 6$, and $mn = 2$ for MM- k NN on this dataset. Similar to the MPEG-7 dataset, we show the traditional Bullseye test scores by retrieving the top 150 leaf instances and we show a modified Bullseye test retrieving the top 75 instances.

As summarized in Table 3, we can see that our method is comparable to LP, LCDP, and MM- k NN for the traditional Bullseye test, and performs better on the modified Bullseye test when retrieving the ground-truth number of shape instances. The DN-TPG performs very well on this dataset and our method can utilize this success when DN-TPG is selected as $d(S_i, S_j)$. Table 4 summarizes the results of testing our method in conjunction with LP, LCDP, MM- k NN, and DN-TPG. It is clear that our method can significantly improve the existing methods and produces the best performance when DN-TPG is selected as $d(S_i, S_j)$.

Method	Top 150	Δ	Top 75	Δ
IDSC + Our Method	96.20 %	6.78 %	91.40 %	14.88 %
IDSC + DN-TPG	98.15 %	8.73 %	92.35 %	15.83 %
IDSC + MM- k NN	96.33 %	6.91 %	88.21 %	11.69 %
IDSC + LCDP	96.44 %	7.02 %	89.46 %	12.94 %
IDSC + LP	95.46 %	6.04 %	90.73 %	14.21 %
IDSC	89.42 %	0.0 %	76.52 %	0.0 %

Table 3. Performance of the proposed method, LP, LCDP, MM- k NN, and DN-TPG methods using the IDSC similarity measure as $d(S_i, S_j)$ on the Swedish Leaf dataset, using the top 75 and top 150 Bullseye evaluation.

Method	Top 150	Δ	Top 75	Δ
DN-TPG + Our Method	99.08 %	0.93 %	94.55 %	2.2 %
MM- k NN + Our Method	98.99 %	2.66 %	91.46 %	3.25 %
LCDP + Our Method	97.80 %	1.36 %	92.63 %	3.17 %
LP + Our Method	96.58 %	1.12 %	90.80 %	0.07 %

Table 4. Performance of the proposed method when applied to the results obtained by LP, LCDP, MM- k NN, or DN-TPG on the Swedish Leaf dataset using IDSC as the base similarity measure.

Nister and Stewenius (N-S) dataset: The N-S dataset is an image dataset containing 10, 200 images. A small subset of these images is illustrated in Fig. 5. The dataset is composed of 2, 550 image classes with only 4 instances per class. More specifically, the N-S dataset contains 2, 550 images of different objects or scenes, each taken from 4 different viewpoints. Due to the small size of each class, the population cues are limited, making this a very challenging dataset for our method.

The image similarities are computed as described in [9, 26], where the image descriptor is a composite of the Hessian-Affine region detector [17] and the SIFT descriptor [16]. The k-means algorithm is used on the sub-sampled image descriptors to obtain the visual vocabulary.

We compare our algorithm to the Contextual Similarity Measure (CDM) [9] and the DN-TPG method. The CDM can considerably improve bag-of-features based image search. Further, CDM uses the ranking of two instances under consideration, when each is selected as a query, to obtain a new similarity measure. This is related to how our method computes k , however our method also takes explicit cues from the global relationship among a population of instances.

The retrieval rate is computed by: (1) for each query, obtain the top 4 (the total number of instances in each class) most similar instances and (2) compute the average of all correctly retrieved instances with respect to each query. Using this test, the best possible retrieval rate is 4. Table 5 summarizes the results where we observe that our method outperforms both the CDM and DN-TPG methods. This shows that our method can still extract population cues and perform very well even when the class size is very small. For this experiment we set $b_u = 5$, $b_l = 1$, and $t = 0.1$.

Baseline [9]	CDM [9]	DN-TPG [26]	Our Method
3.22	3.57	3.61	3.62

Table 5. Performance of the proposed method, CDM, and TN-DPG methods on the N-S image dataset.

6. Conclusion

In this paper, we proposed a robust method capable of working with any ranking algorithm to increase retrieval performance. Encoding population cues into pairwise shape measures allows the proposed method to construct disjoint components consisting of similar shape instances, which provide consensus when retrieving the top matches for a shape instance. Connectivity was established between components which prevents hard constraints on the number of retrieved shapes. In the experiments we show the versatility of this method by comparing it to and applying it in concert with four state-of-the-art methods.

Acknowledgements: This work was supported, in part, by AFOSR FA9550-11-1-0327, NSF IIS-0951754, NSF IIS-1017199, ARL W911NF-10-2-0060, and MURI ARO W911NF-07-1-0185. We would also like to thank Dr. Robert Sharpley and the Interdisciplinary Mathematics Institute for continual support.

References

[1] N. Alajlan, M. S. Kamel, and G. H. Freeman. Geometry-based image retrieval in binary image databases. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(6):1003–1013, 2008.

[2] X. Bai, B. Wang, X. Wang, W. Liu, and Z. Tu. Co-transduction for shape retrieval. In *European Conference on Computer Vision*, pages III: 328–341, 2010.

[3] X. Bai, X. Yang, L. J. Latecki, W. Liu, and Z. Tu. Learning context sensitive shape similarity by graph transduction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2009.

[4] P. Belhumeur, D. Chen, S. Feiner, D. Jacobs, W. Kress, H. Ling, I. Lopez, R. Ramamoorthi, S. Sheorey, S. White, and L. Zhang. Searching the world’s herbaria: A system for visual identification of plant species. In *European Conference on Computer Vision*, pages IV: 116–129, 2008.

[5] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(24):509–522, April 2002.

[6] M. R. Daliri and V. Torre. Robust symbolic representation for shape recognition and retrieval. *Pattern Recognition*, 41(5), 2008.

[7] R. Duda, P. Hart, and D. Stork. *Pattern Classification*. Wiley, 2001.

[8] R. Gopalan and P. Turaga. Articulation-invariant representation of non-planar shapes. In *European Conference on Computer Vision*, pages 1–8, 2011.

[9] H. Jégou, C. Schmid, H. Harzallah, and J. Verbeek. Accurate image search using the contextual dissimilarity measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(1):2–11.

[10] P. Kontschieder, M. Donoser, and H. Bischof. Beyond pairwise shape similarity analysis. In *Asian Conference on Computer Vision*, pages 655–666, 2009.

[11] L. Latecki and R. Lakaemper. Shape similarity measure based on correspondence of visual parts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(10):1–6, October 2000.

[12] L. Lin, K. Zeng, X. Liu, and S. Zhu. Layered graph matching by composite cluster sampling with collaborative and competitive interactions. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1351–1358, 2009.

[13] H. Ling and D. W. Jacobs. Shape classification using the inner-distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(2):286–299, 2007.

[14] H. Ling, X. Yang, and L. Latecki. Balancing deformability and discriminability for shape matching. In *European Conference on Computer Vision*, pages III: 411–424, 2010.

[15] M. Liu, B. Vemuri, S. Amari, and F. Nielsen. Total bregman divergence and its applications to shape retrieval. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3463–3468, 2010.

[16] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal on Computer Vision*, 60:91–110, November 2004.

[17] K. Mikolajczyk and C. Schmid. Scale and affine invariant interest point detectors. *International Journal on Computer Vision*, 60(1):63–86, 2004.

[18] D. Nistér and H. Stewénus. Scalable recognition with a vocabulary tree. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 2161–2168, 2006.

[19] D. Pedronette and R. Torres. Shape retrieval using contour features and distance optimization. In *International Conference on Computer Vision Theory and Applications*, pages 197–202, 2010.

[20] D. Pedronette and R. Torres. Exploiting contextual information for rank aggregation. In *IEEE International Conference on Image Processing*, 2011.

[21] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *Proceedings of the International Conference on Computer Vision*, volume 2, pages 1470–1477, 2003.

[22] O. J. O. Söderkvist. Computer vision classification of leaves from swedish trees. Master’s thesis, Linköping University, SE-581 83 Linköping, Sweden, September 2001. LiTH-ISY-EX-3132.

[23] A. Temlyakov, B. Munsell, J. Waggoner, and S. Wang. Two perceptually motivated strategies for shape classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2289–2296, 2010.

[24] C. Xu, J. Liu, and X. Tang. 2d shape matching by contour flexibility. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(1):180–186, 2009.

[25] X. Yang, S. Koknar Tezel, and L. Latecki. Locally constrained diffusion process on locally densified distance spaces with applications to shape retrieval. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 357–364, 2009.

[26] X. Yang and L. Latecki. Affinity learning on a tensor product graph with applications to shape and image retrieval. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2369–2376, 2011.