Pattern Recognition Letters 83 (2016) 388–394

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Co-spectral for robust shape clustering☆

Song Bai, Zhiyong Liu, Xiang Bai*

School of Electronic Information and Communications, Huazhong University of Science and Technology, 430074 Wuhan, China

ARTICLE INFO

Article history:
Available online 27 July 2016

Keywords:
Shape clustering
Co-training
Shape context
Skeleton path

ABSTRACT

Shape clustering is a difficult visual task due to large intra-class variations and small inter-class variations induced by shape articulation, rotation, occlusion, *etc*. To tackle this problem, we attempt to leverage the complementary nature among features of different statistics (*e.g.*, skeleton-based descriptors and contour-based descriptors) for robust clustering. In this paper, a similarity fusion framework based on spectral analysis is proposed. The proposed method, which we call co-spectral, is a spectral clustering algorithm. It has two inborn merits for shape clustering: (1) it can automatically make use of the complementarity of various shape similarities based on a co-training framework; (2) it does not require shape representations to be vectors. Co-spectral is evaluated on several popular shape benchmarks. The experimental results demonstrate that co-spectral outperforms the state-of-the-art algorithms by a large margin.

1. Introduction

Shape clustering [16] is a fundamental problem in pattern recognition with applications to shape matching [7,18], recognition [13], retrieval [5,6] and classification [4]. Given a collection of shapes $\mathcal{S} = \{s_1, s_2, \ldots, s_N\}$ where N denotes the number of shapes, the aim of shape clustering is to divide all the shape instances into K clusters $\mathcal{C} = \{c_1, c_2, \ldots, c_K\}$ according to a pre-defined similarity measure.

The key issues in shape clustering lie in two aspects. First, shape data is usually not represented by vectorial features. Instead, tree [8], matrix [13,27] and string [17] are more widely-used in shape analysis. Hence, some clustering algorithms that require vectorial representations as inputs, *e.g.*, K-means [28], are not applicable directly. Second, there are large intra-class variations and small inter-class variations, like articulation, rotation, occlusion, *etc*. Nevertheless, it is difficult to design generic and discriminative shape features to handle all the common deformations. In most cases, a certain descriptor only focuses on a specific geometric structure of shapes. For example, Shape Context (SC) [13], as a representative contour-based descriptor, works well with rigid shapes. In contrast, Inner Distance Shape Context (IDSC) [27], which replaces the Euclidean distance used in SC with geodesic distance, is better at dealing with articulated shapes.

To address the above issues, we introduce spectral clustering as an elegant mathematical tool for the shape clustering task. Spectral clustering operates on a weighted affinity graph, where the nodes in the graph represent the data points and the edge weights measure the similarities between two adjacent data points. Therefore, it can deal with arbitrary types of input data, as long as the pairwise similarities are avaliable. This property is crucial for shape clustering, since many shape similarity measures have no vectorial representations. Moreover, by exploiting the properties of Laplacian of the affinity graph, spectral clustering can capture the main patterns across categories and diminish the negative influences of noisy attributes. As a result, spectral clustering is more robust to shape outliers.

Considering the limitation of using only one type of similarity measure, it can be expected that an effective method which integrates multiple complementary similarities can boost the performance of shape clustering remarkably. Nevertheless, it is very difficult to fuse multiple descriptors in shape clustering, since no prior or extra information can be used to judge the discriminative power of different features in such an unsupervised task. To our best knowledge now, no methods have properly addressed the feature fusion issue in the shape clustering task.

In this paper, based on spectral clustering, a co-trained spectral clustering algorithm is presented. The proposed method inherits the nice properties of spectral clustering as introduced above. Similarity fusion is automatically done based on the co-training framework [41] that exploits the complementary nature among different shape descriptors. Moreover, a density-based seed is exerted to co-trained spectral clustering in order to avoid local minima and provide stable performances consistently. At last, since co-training is not guaranteed to converge as suggested in [23], we propose a simple yet effective consensus-based voting scheme to aggregate

☆ This paper has been recommended for acceptance by Gabriella Sanniti di Baja.

* Corresponding author. Fax: +86 87543236.
E-mail address: xbai@hust.edu.cn (X. Bai).

http://dx.doi.org/10.1016/j.patrec.2016.07.014
0167-8655/

the clustering results of different iterations without impairing the performance too much.

The rest of the paper is organized as follows. We give a brief review of shape clustering algorithms in Section 2. Our proposed method is introduced in Section 3. The experimental evaluations and comparisons are conducted in Section 4. Conclusions are drawn in Section 5.

2. Related work

In recent years, many algorithms are proposed to address the shape clustering task. They can be coarsely divided into two categories: contour-based methods and skeleton-based methods.

In [24], a new similarity measure between a single shape and a shape group is defined, and it serves as the basis for a soft K-means like framework to enable robust clustering. Clustering in [35] is achieved by building on a differential geometric representation of shapes and geodesic lengths as shape metrics. Yankov et al. [38] take isomap clustering using a rotationally invariant metric, which can detect the intrinsic nonlinear embedding in which the shape examples reside. In [29], the elastic properties of shape boundaries are investigated and clustering is done using dynamic programming based on the elastic geodesic distance.

Demirci et al. [19] construct a medial axis graph for shape silhouettes. For every two graphs, a many-to-many correspondence between graph nodes [20] is established. These correspondences are used later to recover the abstracted medial axis graph. An information theoretic framework is presented in [37]. It attempts to learn a mixture of tree unions that best accounts for the observed samples using a minimum encoding criterion. In [21], a game theoretic clustering approach is developed, which can simultaneously learn categories from examples and the similarity measures related to them. Shen et al. also propose a skeleton-based clustering algorithm in [33] on the common structure skeleton graph (CSSG), which can discover intrinsic structural information of shapes belonging to the same cluster.

Most aforementioned shape clustering methods are either contour-based or skeleton-based. There are also some methods that are not descriptor-based, such as Laplacian spectrum [30] or minimum spanning trees [40]. The method proposed in this paper is descriptor-based. It combines complementary shape features in a unified framework, thus providing much better performances.

3. Proposed method

3.1. Similarity measure

Contour-based descriptors and skeleton-based descriptors are two main streams in shape analysis. Contour-based descriptors deliver the distribution of shape boundary points. They are more stable to affine transformations. By contrast, skeleton-based descriptors convey the structure of object skeletons, thus are more adequate in non-ridge analysis. The complementary nature between them has been extensively testified in shape recognition [9,32].

Two similarity measures are implemented in this paper, *i.e.*, Shape Context (SC) [13] and Skeleton Path [8], with the former one as a representative contour descriptor and the latter one as a skeleton descriptor.

Shape contetxt. Given a certain shape $s_q \in \mathcal{S}$, we extract its outer contour represented by n discrete points $\nu = \{v_1, v_2, \ldots, v_n\}$ in the plane. Around each point $v_i \in \nu$, we construct a log-polar coordinate space with 12 bins for dividing angle space and 5 bins for dividing radius space. As a consequence, the k-th element in the shape context histogram of v_i is computed via

$$p_i(k) = |\{v | v \in bin(k), v \neq v_i, v \in \nu\}|, \tag{1}$$

where $|.|$ measures the cardinality of the input set.

Let $q = \{q_1, q_2, \ldots, q_n\}$ and $p = \{p_1, p_2, \ldots, p_n\}$ denote two sets of shape context histograms of s_q and s_p respectively. Their pointwise matching cost is measured using χ^2 distance as

$$C(p_i, q_j) = \frac{1}{2}\sum_k \frac{[p_i(k) - q_j(k)]^2}{p_i(k) + q_j(k)}. \tag{2}$$

After obtaining the matching cost $C(p_i, q_j)$ for all pairs of elements $p_i \in p$ and $q_j \in q$, Hungarian algorithm is applied to find the optimal correspondence as

$$H(\pi) = \arg\min_{\pi} \sum_i^n C\left(q_i, p_{\pi(i)}\right), \tag{3}$$

where π is a permutation indicating that the matching is one-to-one.

Skeleton path. We implement skeleton path proposed in [8] as the second similarity measure, which is based on skeleton analysis. In this subsection, we give a brief review of skeleton path. One can refer to the study in [8] for more details if necessary.

Assuming that the skeleton curve is one pixel wide, three kinds of points are defined: end point, junction point and connection point. The end point is defined as the skeleton point owning one adjacent point. The skeleton point with more than two adjacent points is named a junction point, and the rest are connection points. To build the skeleton graph, both end points and junction points are chosen as the nodes of the graph. The edges between two adjacent nodes are the skeleton branches between them.

Given a pair of nodes u, v in a skeleton graph, the skeleton path $P(u, v)$ is defined as the shortest path along the skeleton graph between u and v. Let $P(u_q, v_q)$ and $P(u_p, v_p)$ represent two skeleton paths from shape s_q and s_p respectively. Their path distance is defined as

$$pd(P(u_q, v_q), P(u_p, v_p)) = \sum_{i=1}^{M} \frac{(r_{qi} - r_{pi})^2}{r_{qi} + r_{pi}} + \alpha \frac{(l_q - l_p)^2}{l_q + l_p}, \tag{4}$$

where r_{qi} and $r_{pi} (0 \leq i \leq M)$ represent the radii of the maximal disks centered at the M sample points of skeleton paths $P(u_q, v_q)$ and $P(u_p, v_p)$ respectively. l_q and l_p are their lengths, and α is the weighting factor.

Assume that the skeleton graph of s_q, denoted as $E_q = \{e_{q_1}, \ldots, e_{q_{nq}}\}$, has nq end points, and the skeleton graph of s_p, denoted as $E_p = \{e_{p_1}, \ldots, e_{p_{np}}\}$, has np end points. The pairwise distance between each pair of end points e_{q_i} and e_{p_j}, referred as $ed(e_{q_i}, e_{p_j})$, is computed via Optimal Subsequence Bijection (OSB) as in [8]. Then we can get a $nq \times np$ distance matrix $\Phi(E_q, E_p)$:

$$\begin{pmatrix} ed(e_{q_1}, e_{p_1}) & ed(e_{q_1}, e_{p_2}) & \ldots & ed(e_{q_1}, e_{p_{np}}) \\ ed(e_{q_2}, e_{p_1}) & ed(e_{q_2}, e_{p_2}) & \ldots & ed(e_{q_2}, e_{p_{np}}) \\ \ldots & \ldots & \ldots & \ldots \\ ed(e_{q_{nq}}, e_{p_1}) & ed(e_{q_{nq}}, e_{p_2}) & \ldots & ed(e_{q_{nq}}, e_{p_{np}}) \end{pmatrix} \tag{5}$$

After applying the Hungarian algorithm on $\Phi(E_q, E_p)$, we can get the optimal correspondence $\varphi : \{e_{q_1}, \ldots, e_{q_{nq}}\} \rightarrow \{e_{p_1}, \ldots, e_{p_{np}}\}$ between end points in E_q and those in E_p. Thus the matching cost, also the dissimilarity between two shapes, is obtained.

3.2. Co-trained spectral clustering

As a representative of graph-based clustering algorithms, spectral clustering exploits the properties of Laplacian of the affinity graph. Spectral clustering algorithms are usually divided into two

Algorithm 1 The algorithm of spectral clustering.

Input: The similarity matrix $W \in \mathbb{R}^{N\times N}$
Output: The clustering result $I \in \mathbb{R}^{N\times 1}$.
1: Compute the diagonal matrix $D(i,i) = \sum_j W(i,j)$;
2: Compute the graph Laplacian $l = D^{-1/2}WD^{-1/2}$;
3: Let $U \in \mathbb{R}^{N\times k}$ denote the top-k eigenvectors of l, which are $\mathcal{L}_2$ row-normalized later;
4: Treat each row of U as a feature, and apply standard K-means to produce the final clustering result I.

categories: normalized spectral clustering and unnormalized spectral clustering. What we adopt here is normalized spectral clustering, since it has been extensively testified that normalized Laplacian affinity graph can yield better performances. The pseudocode of spectral clustering used in this paper is given in Algorithm 1.

The primary drawback of spectral clustering is that it can only deal with one type of similarity measure. In order to adapt it to multi-view settings, *i.e.*, multiple independent features are available, a possible solution is leveraging co-training algorithm [41]. Co-training is initially designed for semi-supervised learning, requiring two views of data. The basic assumption of co-training is that either view of data is sufficient to predict the class labels of instances accurately. Co-training is an iterative algorithm. It first learns a separate classifier for each view on labeled data. Then the confident predictions of each classifier on unlabeled data are taken as additional labeled data in the next iteration. To apply co-training to unsupervised shape clustering, a naive way is to change the edge weight of a node pair in one graph according to the clustering results of other graphs. However, it is difficult to determine the degree by which the edge weights are changed.

Our solution here is to project the affinity graph using the eigenvectors of another affinity graph, then back-project it to the original space as in [23]. As a result, the essential information for clustering can be preserved and the intra-clustering information is thrown away. This procedure is conducted conversely and iteratively. The pseudocode of co-trained spectral clustering is given in Algorithm 2. The symmetrization operator on an affinity matrix is defined as $sym(S) = (S + S^{\mathrm{T}})/2$.

Unfortunately, such a co-trained paradigm cannot guarantee convergence. Consequently, if no prior information is available, we cannot determine how many iterations we need and when the best performance is achieved. To tackle the problem, we propose a consensus-based voting scheme in Section 3.4 to aggregate the clustering results of different iterations, thus mining their consensus clusters automatically.

Algorithm 2 The algorithm of co-trained spectral clustering.

Input: Similarity matrices: W_1, W_2
Output: The clustering results $I_k^1, I_k^2 \in \mathbb{R}^{N\times 1}$.
1: Initialization: for $v = 1, 2$, $l_v = D_v^{-1/2}W_vD_v^{-1/2}$, where D_v is a diagonal matrix with $D_v(ii) = \sum_j W_v(i,j)$; Let $U_v^0 \in \mathbb{R}^{N\times k}$ denote the top-k eigenvectors of l_v.
2: **for** $k = 1$ to *ITER* **do**
3: $S_1^k = sym(U_2^{k-1}U_2^{k-1^{\mathrm{T}}}W_1)$;
4: $S_2^k = sym(U_1^{k-1}U_1^{k-1^{\mathrm{T}}}W_2)$;
5: Take S_1^k and S_2^k as the new affinity matrices. Solve for the largest k eigenvectors to obtain U_1^k and U_2^k.
6: Row-normalize U_1^k and U_2^k.
7: Apply standard K-means to produce the clustering result I_k^1 and I_k^2.
8: **end for**

3.3. Density-initialized seeds

In the procedure of spectral clustering, K-means is used to provide the final clustering results. However, it is known that K-means is prone to reach local minima with random initialization of centroids. As a result, two runs of spectral clustering probably result in two entirely different clusters.

To remedy this, we propose an efficient method to initialize the centroids inspired by the recent development of density analysis on the data manifold. Similar to density peak [31], our motivation is that centroids are surrounded by neighbors with lower local density and they are also far away from any points with higher local density.

The local density ρ_i of shape s_i in the feature manifold is

$$\rho_i = \sum_j \mathcal{H}(d_{ij} - d_c), \tag{6}$$

where d_c is a pre-defined cut-off distance, and $\mathcal{H}(\cdot)$ is the indicator function defined as

$$\mathcal{H}(x) = \begin{cases} 1, & if\ x < 0, \\ 0, & otherwise. \end{cases} \tag{7}$$

As suggested in [31], d_c is determined so that the average number of neighbors is 2% of the total number of points in the feature space. Then, we use σ_i to measure the minimum distance between shape s_i and any other shapes with higher density:

$$\sigma_i = \min_{j:\rho_j > \rho_i} (d_{ij}). \tag{8}$$

For the shape with the highest density, we manually take $\sigma_i = \max_j(d_{ij})$.

As a result, our density-initialized seeds are set as follows. For each view and each iteration of co-trained spectral clustering, the low dimensional embedding of all shapes is obtained as spectral clustering usually does. Then, we estimate the local density ρ_i and σ_i for each shape $s_i \in \mathcal{S}$ in the induced embedding space. By sorting the shapes in decreasing order according to $\rho_i\sigma_i$, the top-K shapes are taken as the seeds for centroid initialization when applying the standard K-means.

3.4. Consensus-based voting

Co-trained spectral clustering is an iterative algorithm without guaranteed convergence, hence it should be stopped within a fixed number of iterations. However, without the property of convergence, one cannot know which similarity measure at which iteration yields the best performance if no prior knowledge is available. To address this issue, we propose a novel and robust method that aggregates the clustering results of different similarity measures and different iterations.

We aim at identifying the best-performing clustering result such that it will have the largest confidence score. However, it is the chicken or the egg dilemma here. If the best performance can be identified, we can determine the confidence scores of different clustering results accordingly, and vice versa. Our solution is to mine the consensus information among different clustering results, and vote for the so-called "correct clusters" by using the consensus information.

The co-trained spectral clustering is run for finite iterations (10 times in our experiments). Let $I_k(s_i)$ denote the function that

gives shape s_i clustering assignment in the k-th iteration. Then we define

$$L(s_i, s_j) = \sum_k \delta_{I_k(s_i), I_k(s_j)}, \tag{9}$$

where δ is the Kronecker delta.

In a sense, $L(s_i, s_j)$ can be also interpreted as a new type of affinity between shape s_i and s_j, *i.e.*, if s_i and s_j are classified into the same cluster in most iterations, they are similar with each other. It naturally inspires us to use spectral clustering in Algorithm 1 on L to obtain the final clustering result.

Meanwhile, we can identify those shape pairs (s_i, s_j) that satisfy $L(s_i, s_j) \geq \xi$ as correct pairs. Since we have two similarity measures and the iteration number is 10, 20 clustering results can be obtained. We set ξ to 15 throughout our experiments, which means that if 3/4 clustering results classify a pair of shapes in the same cluster, we identify them to be a correct pair. By doing so, the confidence score of each clustering result can be determined by counting how many times it hits the correct pairs. The weight is determined by applying sigmoid function to the number of correct pairs. Let w_k denote the weight learned in such a voting scheme, then the affinity graph built here is slightly different from Eq. (9) as

$$L_w(s_i, s_j) = \sum_k \delta_{I_k(s_i), I_k(s_j)} w_k. \tag{10}$$

The experimental results suggest that when spectral clustering is applied to Eq. (10) instead of Eq. (9), around 1 percent performance gain in NMI can be achieved.

4. Experiments

To evaluate the effectiveness of the proposed method, three standard shape datasets are chosen for comparison following [33]:

- Aslan and Tari dataset with 56 shapes [2]: it contains 56 shapes, with 4 shapes in each class.
- Aslan and Tari dataset with 180 shapes [1]: it consists of 180 shapes divided into 30 classes, with 6 shapes in each class.
- Aslan and Tari dataset with 1000 shapes [12]: it consists of 1000 articulated shapes, grouped into 50 categories with 20 shapes in each class.

We compare the performance of the proposed co-spectral method against several state-of-the-art algorithms listed as follows:

- CSD [33]: Common Structure Discovery (CSD) is a representative agglomerative hierarchical shape clustering algorithm. It can extract the common structures which capture the intrinsic intra-class structural information of clusters of shapes.
- Game theoretic approach [21]: it is a game theoretic approach that can discover shape categories and compute the corresponding contextual similarities.
- Foreground focus [26]: it first initializes the image-level grouping based on feature correspondences, and refines cluster assignments based on the evolving intra-cluster pattern of local matches iteratively.
- IDSC+Ncuts: Inner Distance Shape Context (IDSC) [27] is used as raw shape descriptors, and Dynamic Programming (DP) is used for pairwise matching under χ^2 metric. Normalized cuts [34] is used for clustering.
- JPD+Ncuts: Normalized cuts with junction path distance [8] as the similarity between shapes.
- JPD+AHC: agglomerative hierarchical clustering with junction path distance as the similarity between shapes.

All the results are quoted from [21,33].

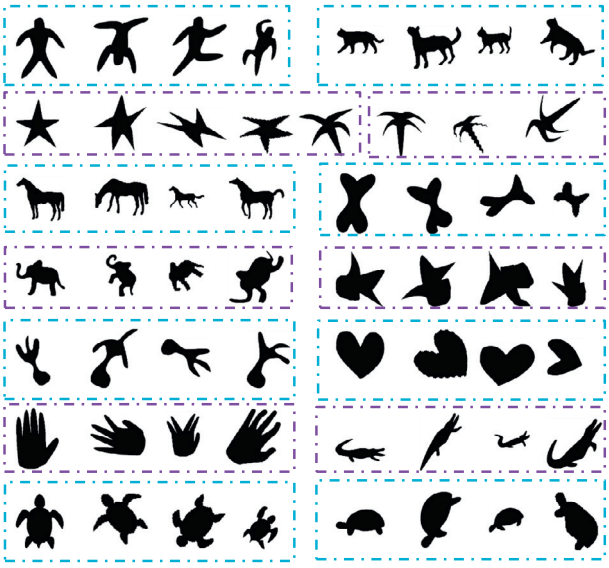

Fig. 1. The clusters on Aslan and Tari dataset with 56 shapes generated by our method.

4.1. Implementation details

If not stated otherwise, we adopt the following setup for all the experiments.

First, we compute pairwise shape distances using shape context and skeleton path. The weight factor α in computing skeleton path is set to 100, since the performance is insensitive to this value as suggested in [33]. Gaussian kernel is applied to build two affinity graphs, serving as two independent views. Then, the two views are fed into co-trained spectral clustering for 10 iterations, incorporated with density-initialized seeds. We manually set the desired number of clusters to be equal to the natural number of classes on each dataset. Finally, consensus-based voting scheme is used to get the clustering result.

The evaluation metric used in this paper is normalized mutual information (NMI), a widely-accepted metric in clustering analysis [21,33]. Our performance comparison involves two parts: qualitative examples and quantitative analysis.

4.2. Qualitative analysis

The clusters generated by co-spectral on Aslan and Tari dataset with 56 shapes are illustrated in Fig. 1. As can be drawn from the figure, our method achieves the nearly perfect performance, and the purity in each cluster is extremely high. Among all the 56 shapes, there is merely one error: one windmill is clustered into pentagrams. This is caused by small inter-class variations between windmill class and pentagram class. Even when two complementary features are used, it is still hard to tell them apart.

Common structure discovery (CSD) achieves the second best performance on this dataset. As illustrated in Fig. 2(a), two errors occur, *i.e.*, four windmills are grouped with pentagrams and a bone forms a unique cluster. The primary reason behind this is that the skeleton-based descriptor used in CSD fails to capture the object silhouette. Whereas, shapes in these two classes are more easily to be distinguished using contour-based descriptor, since they share very similar skeleton structures. In this case, our method benefits from the usage of shape context and significantly improves the discriminative power of the skeleton path.

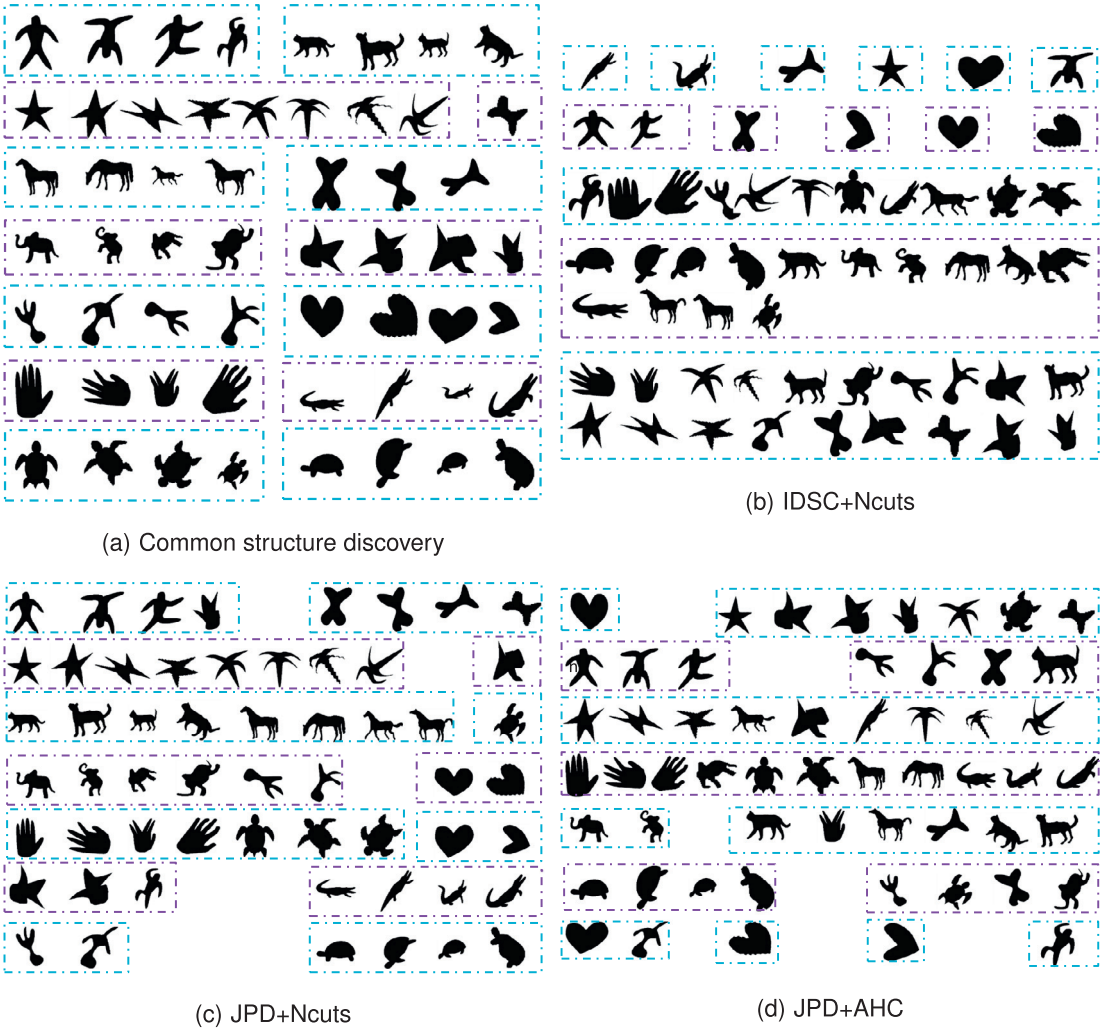

Fig. 2. The clusters on Aslan and Tari dataset with 56 shapes achieved by common structure discovery (a), IDSC+Ncuts (b), JPD+Ncuts (c) and JPD+AHC (d).

Table 1
The performance comparison in NMI of different algorithms.

Methods	56 shapes	180 shapes	1000 shapes
IDSC+Ncuts	0.5660	0.5423	0.5433
JPD+Ncuts	0.6174	0.5785	0.4549
JPD+AHC	0.8674	0.8793	0.7693
Foreground focus	–	–	0.7329
CSD	0.9734	0.9694	0.8096
Game theoretic approach	–	–	0.8722
Shape context+spectral	0.9418	0.9264	0.9676
Skeleton path+spectral	0.9426	0.9746	0.9154
Co-spectral (ours)	**0.9900**	**0.9901**	**0.9711**

4.3. Quantitative analysis

For a numerical quantitative analysis, we use normalized mutual information (NMI) to measure the clustering performance. The NMIs of our method and other state-of-the-art algorithms on the Aslan and Tari dataset with 56 shapes, 180 shapes and 1000 shapes are listed in Table 1.

As we can see from the table, the proposed method achieves the best performance on all datasets. Compared with common structure discovery, co-trained spectral clustering yields 1.60%, 2.07% and 16.15% improvements on Aslan and Tari dataset with 56 shapes, 180 shapes and 1000 shapes respectively. Foreground focus [26] can learn the categories in an unsupervised way through the local matching of features. The result of foreground focus comes from [21], where inner distance shape context [27] is extracted on the boundary points and Normalized cuts [34] is adopted to determine the clusters. Note that co-spectral beats foreground focus by about 20 percents on Aslan and Tari dataset with 1000 shapes.

On the largest Aslan and Tari dataset with 1000 shapes, the previous state-of-the-art performance is achieved by the game theoretic approach [21]. It discovers the shape categories and the corresponding contextual similarities simultaneously. However, its performance is 10 percents inferior to our approach.

The performances of shape context and skeleton path using normalized spectral clustering (Algorithm 1) are also given. We also insert density-initialized seeds to it as co-spectral does. It can be observed that co-spectral achieves better performances than the standard spectral clustering.

4.4. Experiments on MPEG-7 dataset

MPEG-7 dataset [25] consists of 1400 shapes divided into 70 categories. Each category has 20 shapes. Since MPEG-7 dataset is initially designed for shape retrieval, few methods have reported the clustering results. Following [11], we use Inner Distance Shape Context (IDSC) [27] as the input similarity measure. The other similarity measure we choose is based on Shape Context, since the complementary nature between IDSC and SC has been proven

Table 2
The performance comparison in NMI on MPEG-7 dataset.

Methods	SC	IDSC
Spectral clustering (Algorithm 1)	0.8454	0.7813
Graph transduction [11]	0.8600	
Co-spectral (ours)	**0.9301**	

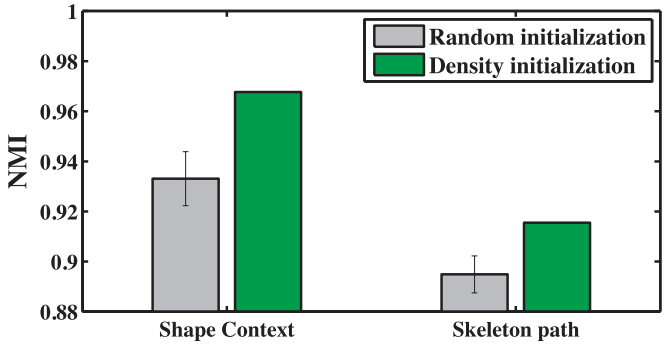

Fig. 3. The advantage of density-initialized seeds.

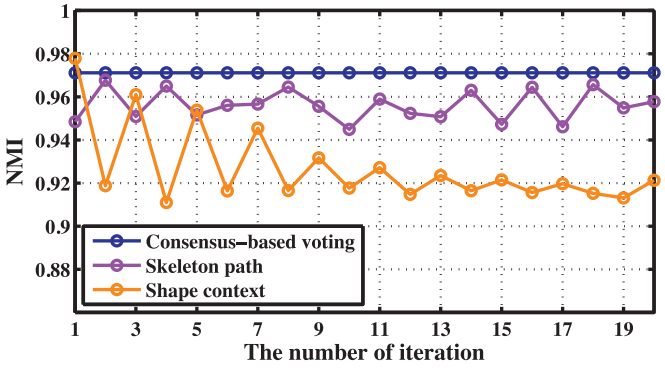

Fig. 4. The performances of each single view counterpart in co-trained spectral clustering, as iteration number increases. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article).

Table 3
The execute time comparison on Aslan and Tari dataset with 56 shapes.

Common structure discovery [33]	Co-spectral (ours)
204.58 s	**0.72 s**

extensively in shape retrieval [3,10]. The evaluation metric used in [11] is defined as the ratio of the number of correct pairs of shapes to the highest possible number of correct pairs. Hence, the best score is 1.

As can be seen from Table 2 clearly, the proposed co-spectral outperforms Graph Transduction (GT) [11] by a large margin. It should be mentioned here that GT is a context-based similarity learning algorithm. It aims at learning a more faithful similarity metric for robust shape retrieval. In order to extend it to shape clustering, affinity propagation [22] is applied to the learned similarity for clustering. Affinity propagation does not require the prior knowledge about the natural number of categories. Thus, the number of clusters produced by affinity propagation is not necessarily equal to the number of categories. As suggested in [11], AP produces 58 clusters, smaller than the category number 70 on the MPEG-7 dataset.

There are some other algorithms which report their clustering results on MPEG-7 dataset. These methods (*e.g.*, [14,15,36,39]) only use a subset of MPEG-7 dataset. Therefore, they are not directly comparable to the method proposed in this paper.

4.5. Discussion

Centroid initialization. Spectral clustering usually initializes the centroids randomly, which tends to reach local minima. In Section 3.3, we propose a robust way for centroid initialization using density analysis. In Fig. 3, the advantage of density-initialized seeds over random initialization is given. We apply spectral clustering on two affinity graphs obtained by shape context and skeleton path respectively. To obtain the performance of random initialization, we repeat the clustering process for 50 times and report the average NMI as well as the standard deviation. As can be seen from the figure, density-initialized seeds achieve much better and more stable performances than random initialization.

Consensus voting. In Fig. 4, we plot performances of the single view counterpart of co-trained spectral clustering with density-initialized seeds. As can be seen clearly, co-trained spectral clustering cannot guarantee the convergence of iteration. In fact, most algorithms based on co-training are not convergent at all.

The blue line depicts the performance of consensus-based voting. As we can see, such a voting scheme can well capture the consensus information among the clustering results of different iterations and different views, thus yielding stable and nearly perfect performances.

Complexity analysis. In each iteration, we need to compute two new similarity matrices by multiplying three matrices, whose sizes are $N \times k$, $k \times N$ and $N \times N$ respectively. Recall that k is the embedding dimension. Therefore, the time complexity of our method is upper-bounded by $O(N^3)$, which seems to be a bit high. However, the operation of matrix multiplication can be well optimized using the Coppersmith-Winograd algorithm or Optimized CW-like algorithms.

In Table 3, we give a comparison in execute time on Aslan and Tari dataset with 56 shapes with the state-of-the-art algorithm, *i.e.*, Common Structure Discovery (CSD). CSD is implemented using the public available codes.[1] In order to keep the comparison fair, all the experiments are run with Matlab in a personal computer with an Intel(R) Core(TM) i5 CPU (3.40 GHz) and 16GB memory. As we can see from Table 3, the proposed co-spectral is at least 3 orders of magnitude faster than CSD.

5. Conclusion

In this paper, we address the shape clustering issue based on spectral clustering, taking advantage of its suitable characteristics for shape analysis. Two complementary shape features, *i.e.*, shape context and skeleton path, are fused in the framework of co-trained spectral clustering. We further propose two important improvements over co-trained spectral clustering. Incorporated with density-initialized seeds, spectral clustering can avoid local minima and generate more stable clustering results. Meanwhile, consensus-based voting solves the bottleneck of misconvergence of co-training algorithms, and mines the consensus information among different clustering results. Extensive experimental results demonstrate the advantage of the proposed method over other state-of-the-art algorithms on three popular shape datasets.

[1] http://wei-shen.weebly.com/publications.html.

Acknowledgment

This work was supported in part by NSFC 61573160 and China Scholarship Council.

References

[1] C. Aslan, A. Erdem, E. Erdem, S. Tari, Disconnected skeleton: shape at its absolute scale, Trans. Pattern Anal. Mach. Intell. 30 (12) (2008) 2188–2203.
[2] C. Aslan, S. Tari, An axis-based representation for recognition, in: Proceedings of International Conference on Computer Vision, ICCV, 2005.
[3] S. Bai, X. Bai, Sparse contextual activation for efficient visual re-ranking, Trans. Image Process. 25 (3) (2016) 1056–1069.
[4] S. Bai, X. Bai, W. Liu, Multiple stage residual model for image classification and vector compression, Trans. Multimed. 18 (7) (2016) 1351–1362.
[5] S. Bai, X. Bai, W. Liu, F. Roli, Neural shape codes for 3d model retrieval, Pattern Recognit. Lett. 65 (2015) 15–21.
[6] S. Bai, X. Bai, Z. Zhou, Z. Zhang, L.J. Latecki, Gift: a real-time and scalable 3d shape search engine, in: Proceedings of Computer Vision and Pattern Recognition, CVPR, 2016a.
[7] X. Bai, S. Bai, Z. Zhu, L.J. Latecki, 3d shape matching via two layer coding, Trans. Pattern Anal. Mach. Intell. 37 (12) (2015) 2361–2373.
[8] X. Bai, L.J. Latecki, Path similarity skeleton graph matching, Trans. Pattern Anal. Mach. Intell. 30 (7) (2008) 1282–1292.
[9] X. Bai, W. Liu, Z. Tu, Integrating contour and skeleton for shape classification, in: Proceedings of IEEE Workshop on NORDIA, 2009.
[10] X. Bai, B. Wang, C. Yao, W. Liu, Z. Tu, Co-transduction for shape retrieval, Trans. Image Process. 21 (5) (2012) 2747–2757.
[11] X. Bai, X. Yang, L.J. Latecki, W. Liu, Z. Tu, Learning context-sensitive shape similarity by graph transduction, Trans. Pattern Anal. Mach. Intell. 32 (5) (2010) 861–874.
[12] E. Baseski, A. Erdem, S. Tari, Dissimilarity between two skeletal trees in a context, Pattern Recognit. 42 (3) (2009) 370–385.
[13] S. Belongie, J. Malik, J. Puzicha, Shape matching and object recognition using shape contexts, Trans. Pattern Anal. Mach. Intell. 24 (4) (2002) 509–522.
[14] M. Bicego, V. Murino, Investigating hidden Markov models' capabilities in 2d shape classification, Trans. Pattern Anal. Mach. Intell. 26 (2) (2004) 281–286.
[15] M. Bicego, V. Murino, M.A. Figueiredo, Similarity-based classification of sequences using hidden Markov models, Pattern Recognit. 37 (12) (2004) 2281–2291.
[16] H. Chang, D.-Y. Yeung, Robust path-based spectral clustering, Pattern Recognit. 41 (1) (2008) 191–203.
[17] M.R. Daliri, V. Torre, Robust symbolic representation for shape recognition and retrieval, Pattern Recognit. 41 (5) (2008) 1782–1798.
[18] M.F. Demirci, Y. Osmanlioglu, A. Shokoufandeh, S. Dickinson, Efficient many-to-many feature matching under the l1 norm, Comput. Vis. Image Underst. 115 (7) (2011) 976–983.
[19] M.F. Demirci, A. Shokoufandeh, S.J. Dickinson, Skeletal shape abstraction from examples, Trans. Pattern Anal. Mach. Intell. 31 (5) (2009) 944–952.
[20] M.F. Demirci, A. Shokoufandeh, Y. Keselman, L. Bretzner, S. Dickinson, Object recognition as many-to-many feature matching, Int. J. Comput. Vis. 69 (2) (2006) 203–222.
[21] A. Erdem, A. Torsello, A game theoretic approach to learning shape categories and contextual similarities, Structural, Syntactic, and Statistical Pattern Recognition, Springer, 2010.
[22] B.J. Frey, D. Dueck, Clustering by passing messages between data points, Science 315 (5814) (2007) 972–976.
[23] A. Kumar, H. Daumé, A co-training approach for multi-view spectral clustering, in: Proceedings of International Conference on Machine Learning, ICML, 2011, pp. 393–400.
[24] R. Lakaemper, J. Zeng, A context dependent distance measure for shape clustering, Advances in Visual Computing, Springer, 2008, pp. 145–156.
[25] L.J. Latecki, R. Lakämper, U. Eckhardt, Shape descriptors for non-rigid shapes with a single closed contour, in: Proceedings of Computer Vision and Pattern Recognition, CVPR, vol. 1, 2000, pp. 424–429.
[26] Y.J. Lee, K. Grauman, Foreground focus: unsupervised learning from partially matching images, Int. J. Comput. Vis. 85 (2) (2009) 143–166.
[27] H. Ling, D.W. Jacobs, Shape classification using the inner-distance, Trans. Pattern Anal. Mach. Intell. 29 (2) (2007) 286–299.
[28] S. Lloyd, Least squares quantization in pcm, IEEE Trans. Inf. Theory 28 (2) (1982) 129–137.
[29] W. Mio, A. Srivastava, S. Joshi, On shape of plane elastic curves, Int. J. Comput. Vis. 73 (3) (2007) 307–324.
[30] H.-F. Pan, D. Liang, J. Tang, N. Wang, Shape representation and clustering based on Laplacian spectrum, in: Proceedings of International Congress on Image and Signal Processingn, IEEE, 2009, pp. 1–4.
[31] A. Rodriguez, A. Laio, Clustering by fast search and find of density peaks, Science 344 (6191) (2014) 1492–1496.
[32] W. Shen, Y. Jiang, W. Gao, D. Zeng, X. Wang, Shape recognition by bag of skeleton-associated contour parts, Pattern Recognit. Lett. (2016).
[33] W. Shen, Y. Wang, X. Bai, H. Wang, L.J. Latecki, Shape clustering: common structure discovery, Pattern Recognit. 46 (2) (2013) 539–550.
[34] J. Shi, J. Malik, Normalized cuts and image segmentation, Trans. Pattern Anal. Mach. Intell. 22 (8) (2000) 888–905.
[35] A. Srivastava, S.H. Joshi, W. Mio, X. Liu, Statistical shape analysis: clustering, learning, and testing, Trans. Pattern Anal. Mach. Intell. 27 (4) (2005) 590–602.
[36] N. Thakoor, J. Gao, S. Jung, Hidden Markov model-based weighted likelihood discriminant for 2-d shape classification, Trans. Image Process. 16 (11) (2007) 2707–2719.
[37] A. Torsello, E.R. Hancock, Learning shape-classes using a mixture of tree-unions, Trans. Pattern Anal. Mach. Intell. 28 (6) (2006) 954–967.
[38] D. Yankov, E. Keogh, Manifold clustering of shapes, in: Proceedings of International Conference on Data Mining, ICDM, 2006, pp. 1167–1171.
[39] Z. Zhang, D. Pati, A. Srivastava, Bayesian clustering of shapes of curves, J. Stat. Plan. Inference 166 (2015) 171–186.
[40] C. Zhong, D. Miao, R. Wang, A graph-theoretical clustering method based on two rounds of minimum spanning trees, Pattern Recognit. 43 (3) (2010) 752–766.
[41] Z.-H. Zhou, M. Li, Semi-supervised regression with co-training, in: Proceedings of International Joint Conference on Artificial Intelligence, IJCAI, vol. 5, 2005, pp. 908–913.