Proceedings Paper:
Bai, L., Cui, Lixin, Wang, Yue orcid.org/0000-0002-2482-005X et al. (3 more authors) (2016) Shape classification with a vertex clustering graph kernel. In: Davis, L., Bimbo, A. Del and Lovell, B., (eds.) 2016 23rd International Conference on Pattern Recognition (ICPR). IEEE Computer Society, Los Alamitos, CA, USA, pp. 2634-2639.

https://doi.org/10.1109/ICPR.2016.7900032

Shape Classification with A Vertex Clustering Graph Kernel

Lu Bai*, Lixin Cui*, Yue Wang*, Xiao Bai†, and Edwin R. Hancock‡
*School of Information, Central University of Finance and Economics, 39 South College Road, Beijing, China
†School of Computer Science and Engineering, Beihang University, Beijing, China
‡Department of Computer Science, University of York, York, YO10 5DD, UK

***Abstract*—Graph kernels are powerful tools for structural analysis in computer vision. Unfortunately, most existing state-of-the-art graph kernels ignore the locational or structural correspondence information between graphs, based on the visual background. This drawback influences the performance of existing kernels for computer vision based classification problems, e.g., classification of shapes, point clouds and digital images. The aim of this paper is to address the problem with existing kernels, by developing a novel vertex clustering graph kernel. We show that this kernel not only overcomes the shortcoming of ignoring correspondence information between isomorphic substructures that arises in most existing graph kernels, but also guarantees the transitivity between the correspondence information. Our kernel can easily outperform state-of-the-art graph kernels in terms of classification accuracy on standard shape based graph datasets.**

I. Introduction

Graph structures are important tools for representing computer vision data (e.g., 3D shapes [1], point clouds [2], digital images [3] and videos [4]), since they can reflect structural and relational arrangements of objects in a scene. One challenge arising in classifying graph-based computer vision data is how to accurately and effectively compute the graph similarities for classification. One way to address this problem is to use graph kernels.

A. Graph Kernels on Computer Vision

Graph kernels have been proven powerful tools for structural analysis in computer vision. Typical applications include a) image classification [5], b) 3D shape classification [1], c) handwriting recognition [6], and d) point cloud classification [1]. The main advantages of using graph kernels are twofold. First, graph kernels can characterize graph features in a high dimensional space and thus have the capability of preserving graph structures. Second, graph kernels make the rapidly developing kernel machinery for vectorial data applicable to graphs.

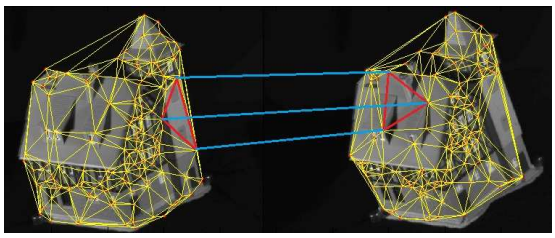


Fig. 1. Two graphs abstracted from two digital images containing the same object, based on different viewpoints.

One of the most successful and widely used approach to defining kernels between a pair of graphs is to decompose the graphs into substructures and to compare/count pairs of specific isomorphic substructures. Specifically, all available graph decomposition methods can be used to define a graph kernel, e.g., graph kernels based on counting pairs of isomorphic a) walks [7], b) paths [8], and c) restricted subgraph or subtree substructures [9]. With this scenario, Gaidon et al. [10] have developed a subtree kernel for comparing videos. For each video, the method considers complex actions as decomposed spatio-temporal parts and builds corresponding binary trees. The resulting kernel is computed by counting the number of isomorphic subtree patterns. Bach [11] has proposed a family of kernels for comparing point clouds. These kernels are based on a newly developed local tree-walk kernel between subtrees, that is defined by a factorization on the properly defined graphical models of the subtrees. Wang and Sahbi [12] have defined a graph kernel for action recognition. They first describe actions in the videos using directed acyclic graphs (DAGs). The resulting kernel is defined as an extending random walk kernel by counting the number of isomorphic walks of DAGs. Harchaoui and Bath [5] have proposed a segmentation graph kernel for image classification. For this method, each image is represented by a segmentation graph, each vertex corresponds to a segmented region, and each edge joins a pair of neighboring regions. The resulting kernel is computed by counting the inexact isomorphic subtree patterns between segmentation graphs. Other state-of-the-art graph kernels include a) the shortest path graph kernel [13], b) the backtrackless walk kernel [14], c) the Lovás kernel [15], d) the Weisfeiler-Lehman subtree kernel [16], [17], e) the continuous-attribute scalable graph kernel [18], f) the subgraph matching kernel [17], etc. Some kernels have also been used for computer vision applications.

One main drawback arising in the above mentioned kernels is that they do not establish reliable correspondence information between isomorphic substructures. In other words, for graphs abstracted from images or 3D shapes, most existing kernels cannot identify whether the isomorphic substructures are located in identical regions based on the visual background. For an instance, in Fig.1 there are two graphs abstracted from two digital images, both containing the same house object, based on different viewpoints. For these graphs, most existing kernels will identify the two isomorphic subgraphs consisting of red lines and thus contribute an unit value to the kernel. Even though the substructures are not locationally or structurally aligned based on the image background, since the vertices connected by blue lines are not aligned. As a result,

Preprint submitted to 23rd International Conference on Pattern Recognition.
Received April 17, 2016.

this drawback influences the performance of most existing graph kernels for computer vision applications. To overcome the shortcoming, Bai et al. [19] have proposed an aligned subtree kernel. The kernel is computed by counting the number of isomorphic subtrees that rooted at aligned vertices, and thus overcomes the shortcoming of neglecting locational/structural correspondences between isomorphic substructures arising in most graph kernels. Experiments demonstrate that the effectiveness of the aligned subtree kernel on 3D shape classification problems.

Unfortunately, the aligned subtree kernel cannot guarantee the transitivity between aligned vertices. More specifically, given three vertices v, u and w, if v and u are aligned, and u and w are aligned, the kernel cannot guarantee that v and w are also aligned. On the other hand, Fröhlich et al. [20] demonstrate that the transitive alignment step is necessary to guarantee the positive definiteness of vertex alignment kernels. Thus, the aligned subtree kernel cannot be guaranteed as a positive definite kernel. Moreover, all the mentioned kernels only reflect graph characteristics for each pair of graphs under comparisons, and thus ignore the information from other graphs. These drawbacks limit the precision of kernel-based similarity measures. Therefore, developing effective graph kernels still remains an open challenge.

B. Contributions

The aim of this paper is to address the mentioned problems arising in existing graph kernels, by proposing a new vertex clustering graph kernel. To this end, we commence by computing a S-dimensional signature for each vertex as the point coordinate, through the frequencies of the shortest paths rooted from the vertex to the remaining vertices. The signature can represent a vertex in a high dimensional principle space, and thus better preserve the information residing on the vertex (note that any existing vectorial vertex signature can be employed for the framework proposed in this paper). For a set of N graphs under comparisons, we perform the K-means clustering method [21] on their vertex signatures and divide the vertices into N clusters. More formally, we select N centroid points and minimize the sum of the squared distances from each vertex to its nearest centroid point. Since a cluster of vertices are all aligned to the corresponding centroid point, these vertices can be seen as being approximately aligned. The resulting kernel for a pair of graphs is computed by counting the number of aligned vertex pairs. We show that this kernel not only overcomes the shortcoming of neglecting substructure correspondences, but also guarantees the transitivity between the correspondence information. Moreover, the computation of the new kernel for a pair of graphs encapsulates the information from other graphs, and thus reflects richer graph characteristics. Experiments demonstrate that our kernel can outperform state-of-the-art graph kernels on shape-based computer vision classification problems.

II. Preliminary Concepts

A. Vertex Signatures from the Shortest Paths

For an undirected graph $G(V, E)$, V is the vertex set and E is the edge set. We commence by computing the shortest path matrix $\mathcal{S}_G$, where $\mathcal{S}_G(v, u)$ records the length of the shortest path between the vertices $v \in V$ and $u \in V$. Let C_v^s be a $|V|$-dimensional vector and satisfies

$$C_v^s(u) = \begin{cases} 1 & \text{if } \mathcal{S}_G(v,u) = s; \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

For the graph $G(V, E)$, we compute a S-dimensional feature vector for the vertex $v \in V$ as

$$F_v^S = [\sum_{u\in V} C_v^1(u), \ldots, \sum_{u\in V} C_v^l(u), \ldots, \sum_{u\in V} C_v^S(u)]^\top, \quad (2)$$

where $l \leq S$, $\sum_{u\in V} C_v^l(u)$ is the l-th element and counts the number of the shortest paths of length l from v to the remaining vertices. If $L_{\max}$ is the greatest length of the shortest paths from v to the remaining vertices and $L_{\max} < l \leq S$, then $C_v^l(u)$ is 0. Note that, for some instances, if $l \leq L_{\max}$, the non-zero l-th elements $\sum_{u_p\in V_p} C_{v_p}^l(u_p)$ and $\sum_{u_q\in V_q} C_{v_q}^l(u_q)$ for the feature vectors $F_{v_p}^S$ and $F_{v_q}^S$ of a pair of graphs G_p and G_q may be the same. This will influence the distinguishable ability of the vertex feature vectors. To overcome this problem, we normalize the feature vector F_v^S and divide each element by the sum of all elements. Thus, we compute a normalized S-dimensional signature $\mathcal{F}_v^S$ for v as

$$\mathcal{F}_v^S = F_v^S / \sum_{v\in V} F_v^S, \quad (3)$$

where the l-th element of $\mathcal{F}_v^S$ is $\sum_{u\in V} C_v^l(u) / \sum_{v\in V} F_v^S$. $\mathcal{F}_v^S$ represents v in a S-dimensional principle space, and can be seen as the point coordinate in the space. Moreover, $\mathcal{F}_v^S$ encapsulates the distribution in terms of the frequencies of different shortest path lengths. As a result, $\mathcal{F}_v^S$ also reflects rich structural information in terms of the shortest paths.

B. Vertex Alignments through K-means Methods

In this subsection, we divide the vertices of graphs into corresponding clusters using the K-means clustering method [21], and identify the alignment between vertices based on the vertex clusters. Let $\mathbf{G} = \{G_1, \ldots, G_p, \ldots, G_q, \ldots, G_N\}$ be a set of N graphs under comparisons. For each graph in $\mathbf{G}$, we compute the S-dimensional signatures of its vertices as points, based on Eq.(3). Assume we have n vertices for the graphs in $\mathbf{G}$, the S-dimensional signatures of the n vertices are $\mathcal{F}^S = (\mathcal{F}_1^S, \mathcal{F}_2^S, \ldots, \mathcal{F}_n^S)$. Given N clusters $\Omega = (y_1^S, y_2^S, \ldots, y_N^S)$ where S corresponds to the parameter of these S-dimensional signatures, the K-means method aims to minimize the sum of square distances (i.e., the Euclidean distance in this paper) between the vertex point $\mathcal{F}_j^S$ and the cluster centroid point of cluster y_i^S to which $\mathcal{F}_j^S$ has been assigned, i.e.,

$$\arg\min_{\Omega} \sum_{i=1}^{N} \sum_{\mathcal{F}_j^S \in y_i^S} \|\mathcal{F}_j^S - \mu_i^S\|^2, \quad (4)$$

where μ_i^S is the mean (i.e., the centroid point) of the vertices in cluster y_i^S in terms of the S-dimensional signatures. Note that, the K-means method requires initial means for the expected clusters. To eliminate the randomness of setting the initial means, we proposed to compute the vertex mean of each graph as the initial N centroid points. For each graph $G_p(V_p, E_p)$, the centroid point μ_p^S of cluster y_p^S is

$$\mu_p^S = \sum_{v_p\in V_p} \mathcal{F}_{v_p}^S / |V_p|. \quad (5)$$

Preprint submitted to 23rd International Conference on Pattern Recognition.
Received April 17, 2016.

Vertices belonging to the same cluster are all mostly closed to the same centroid point, thus these vertices can be seen approximately aligned. Furthermore, the vertex clustering relies on the minimization of Eq.(4), based on the S-dimensional signatures of all graphs. As a result, the clustering results of the vertices encapsulate rich information over all graphs.

III. Graph Kernels from Vertex Clustering

A. The Vertex Clustering Graph Kernel

Let $\mathbf{G} = \{G_1, \ldots, G_p, \ldots, G_q, \ldots, G_N\}$ be a set of N graphs. We commence by computing the S-dimensional signatures of vertices for each graph, based on Eq.(3). We cluster the vertices (i.e., the S-dimensional signatures) of the graphs in $\mathbf{G}$ into N clusters $\Omega = (y_1^S, \ldots, y_j^S, \ldots, y_N^S)$, based on the definition in Section II-B. For a pair of graphs $G_p(V_p, E_p)$ and $G_q(V_q, E_q)$ from $\mathbf{G}$, the vertex clustering graph kernel $\mathcal{K}$ is

$$\mathcal{K}(G_p, G_q) = \sum_{S=1}^{S_{\max}} \sum_{v_p \in V_p} \sum_{v_q \in V_q} \delta^{(S)}(v_p, v_q), \qquad (6)$$

where

$$\delta^{(S)}(v_p, v_q) = \begin{cases} 1 & \text{if } v_p \in y_j^S \text{ and } v_q \in y_j^S, \text{ and} \\ & \mathcal{F}_{v_p}^S(S) \neq 0 \text{ and } \mathcal{F}_{v_q}^S(S) \neq 0; \\ 0 & \text{otherwise.} \end{cases} \qquad (7)$$

For $v_p \in V_p$, $\mathcal{F}_{v_p}^S(S)$ is the S-th element of its S-dimensional signature. $\mathcal{F}_{v_p}^S(S) \neq 0$ guarantees that there are the shortest paths of length S from v_p to the remaining vertices in G_p. The parameter $S_{\max}$ corresponds to the maximum value of the parameter S. Generally, we select the greatest value of the shortest paths of graphs in $\mathbf{G}$ as $S_{\max}$. Clearly, Eq.(6) and Eq.(7) indicate that the proposed graph kernel $\mathcal{K}(G_p, G_q)$ counts the number of vertex pairs that belong to the same cluster y_j^S. As we have stated in Section II-B, the vertices belonging to the same cluster can be seen approximately aligned. As a result, $\mathcal{K}(G_p, G_q)$ can also be seen as a matching kernel that counts the number of aligned or matched vertex pairs for a pair of graphs, i.e., $\mathcal{K}(G_p, G_q)$ establishes the reliable correspondences between vertices.

Time Complexity For a set of N graphs each of which has n vertices, computing the vertex clustering kernel for all pairs of graphs requires time complexity $O(Nn^3 + N^2 S_{\max} n^2 + N^2 nT)$, and T is the iteration number for the K-means method. This is because computing the S-dimensional signatures of vertices for each graph relies on computing the shortest path matrix, and requires time complexity $O(n^3)$. Computing the N means for the S-dimensional signatures of all graphs through the K-means method requires time complexity $O(N^2 nT)$, since there are Nn S-dimensional signature vectors and N means. Computing the kernel values for all pairs of graphs over $S_{\max}$ families of S-dimensional signatures requires time complexity $O(N^2 S_{\max} n^2)$. Thus, the whole time complexity is $O(Nn^3 + N^2 S_{\max} n^2 + N^2 nT)$, our kernel can be computed in polynomial time. Note that we use the fastest K-means MATLAB implementation developed by Deng Cai [22], and the default number of T is 100.

B. Advantages of the Vertex Clustering Kernel

The new vertex clustering kernel has following advantages, that are not available to the mentioned state-of-the-art graph kernels in Section I-A. First, unlike aligned subtree kernel [19] that also identifies the vertex correspondences between each pair of graphs, the proposed kernel $\mathcal{K}$ identifies the vertex correspondences by evaluating whether the vertices are assigned to the same cluster, i.e., we evaluate whether the vertices are mostly closed to or aligned to the same corresponding centroid points. Thus, $\mathcal{K}$ can guarantee the transitivity between pairs of aligned vertices (i.e., for three vertices v, u and w, if v and u are aligned, and u and w are aligned, then v and w are also aligned), and thus are **positive definite** (***pd***). By contrast, the aligned subtree kernel cannot guarantee the transitivity and the positive definiteness properties. Second, unlike all mentioned state-of-the-art graph kernels in Section I-A, that only capture graph characteristics for each pair of graphs under comparisons (i.e., these kernels ignore the information from other graphs), the computation of the proposed kernel $\mathcal{K}$ for each pair of graphs also incorporates the information from other graphs under comparisons. This is because assigning each vertex into a corresponding cluster relies on the minimization of the sum of square distances between all vertices and the centroid points of their clusters. This can be observed by Eq.(4). As a result, $\mathcal{K}$ reflects richer graph characteristics than existing graph kernels.

IV. Experimental Results

We test our kernel on standard computer vision datasets.

BAR31, BSPHERE31 and GEOD31: The SHREC 3D Shape database consists of 15 classes and 20 individuals per class, that is 300 shapes [23]. This is an usual benchmark in 3D shape recognition and plays an important role for classification problems. The original 3D shapes of the SHREC database can be found in Fig.2, where each row corresponds to a class of shapes. It is clear that some shapes from the same class are quite different, the shapes in the SHREC database are hard for classifications. From the SHREC 3D Shape database, three graph datasets, namely BAR31, BSPHERE31 and GEOD31 datasets, are established through three mapping functions. These functions are a) ERG barycenter: distance from the center of mass/barycenter, b) ERG bsphere: distance from the center of the sphere that circumscribes the object, and c) ERG integral geodesic: the average of the geodesic distances to the all other points. The number of maximum, minimum and average vertices for the three datasets are a) 220, 41 and 95.42 (for BAR31), b) 227, 43 and 99.83 (for BSPHERE31), and c) 380, 29 and 57.42 (for GEOD31), respectively.

GatorBait: GatorBait has 100 shapes representing fishes from 30 different classes. The Delaunay graphs have been extracted from their shape quantization (Canny algorithm followed by contour decimation). Since the classes are associated to fish genus and not to species, there is a high intraclass variability in many cases. The graphs of the GatorBait dataset are shown in Fig.3. There are 10 classes with one species, 11 with 1 to 3 individuals, 5 with 4 to 6 individuals and only 4 classes with more than 6 species. Thus, the database, though having only 100 samples, plays a challenging role in testing graph classification. The number of maximum, minimum and average vertices for the dataset are 545, 239 and 348.70.

Preprint submitted to 23rd International Conference on Pattern Recognition.
Received April 17, 2016.

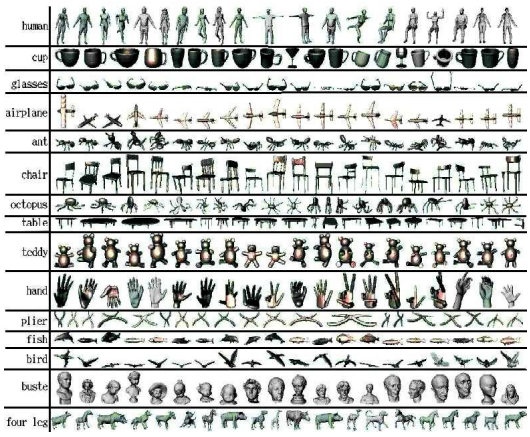


Fig. 2. Examples of the 3D shapes for the Reeb graphs.

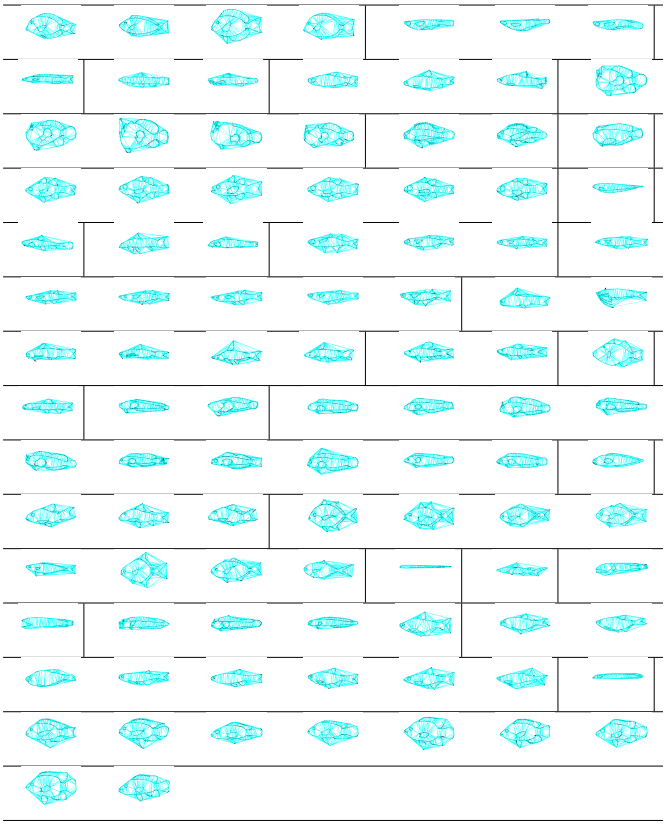
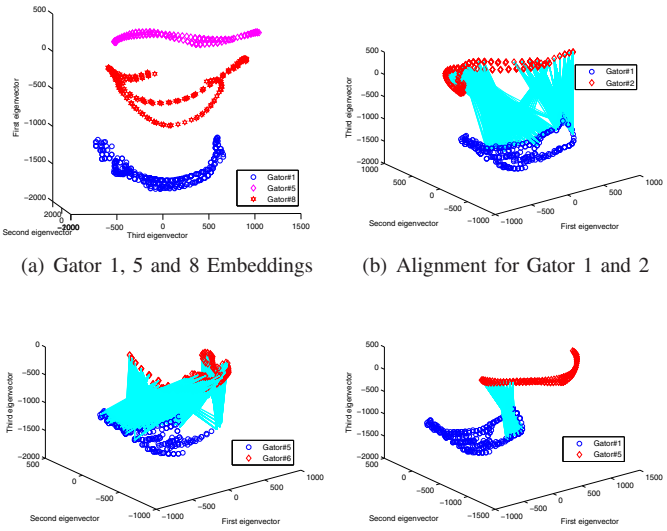


Fig. 3. Examples of the graphs in GatorBait.

Shock: This dataset consists of 150 graphs from the Shock 2D shape database [24]. Each graph is a skeletal representation of the differential structure of the boundary of a shape. There are 10 graph classes, each containing 15 graphs. The number of maximum, minimum and average vertices for the dataset are 33, 3 and 13.16.



(a) Gator 1, 5 and 8 Embeddings (b) Alignment for Gator 1 and 2

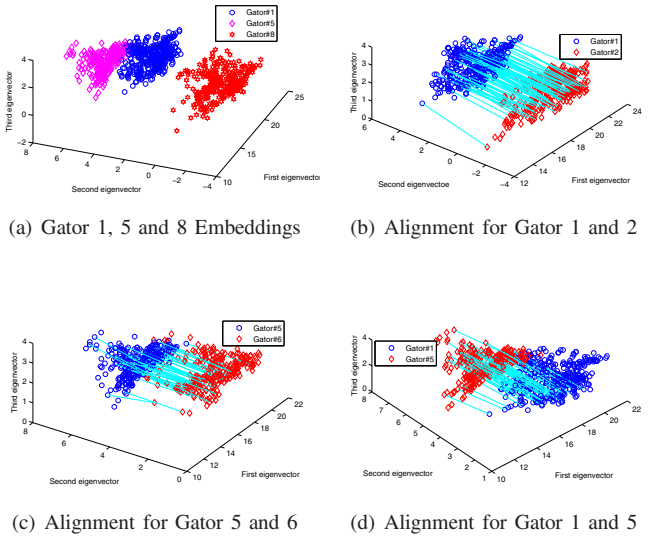
(c) Alignment for Gator 5 and 6 (d) Alignment for Gator 1 and 5

Fig. 4. Embedding and alignment evaluations for VCGK.

A. *Vertex Embeddings and Alignments*

In this subsection, we evaluate the performance of the proposed vertex clustering graph kernel (VCGK) on vertex embedding and alignment problems, through the challenging GatorBait graphs. Moreover, we also compare the VCGK kernel to the aligned subtree kernel (ASK) [19] that also identifies vertex correspondences associated with vectorial signatures of vertices. **a)** For the vertex embedding evaluation, we select the first, fifth and eighth graph structures from the GatorBait dataset, which is shown in Fig.3, as the testing graphs. Based on Fig.3, the three graphs belong to 3 different classes, respectively. For the three graphs, we compute the S-dimensional signatures (for the VCGK kernel) and h-layer entropic representations (for the ASK kernel) as the vectorial representations of vertices. We perform the Principle Component Analysis (PCA) [25] on the vertex representations and embed them into a 3-dimensional principal space. We visualize the embedding results of the vertices using the first three principal components. The embedding results for the VCGK and ASK kernels are shown in Fig.4-a and Fig.5-a, respectively. **b)** For the vertex alignment problems, we visualize the embedding result of vertices of a pair of graphs through the PCA method for each of the VCGK and ASK kernels. Moreover, we visualize the alignment results between vertices for a pair of graphs using cyan lines, i.e., we connect a pair of aligned vertices identified by the VCGK and ASK kernels using the cyan lines. For the graph structures shown in Fig.3, the alignment results identified by the proposed VCGK kernel between the first and second graphs, the fifth and sixth graphs, and the first and fifth graphs are shown in Fig.4-b, Fig.4-c and Fig.4-d, respectively. The alignment results identified by the ASK kernel between the first and second graphs, the fifth and sixth graphs, and the first and fifth graphs are shown in Fig.5-b, Fig.5-c and Fig.5-d, respectively. Note that, based on Fig.3, the first and second graphs are both from the first class of GatorBait, the fifth and sixth graphs are both from the second class of GatorBait.

Preprint submitted to 23rd International Conference on Pattern Recognition.
Received April 17, 2016.



(a) Gator 1, 5 and 8 Embeddings (b) Alignment for Gator 1 and 2

(c) Alignment for Gator 5 and 6 (d) Alignment for Gator 1 and 5

Fig. 5. Embedding and alignment evaluations for ASK.

In terms of the vertex embedding results, Fig.4-a and Fig.5-a indicate that the vertices from the same graph structure tend to aggregate to the same cluster. This indicates that the required S-dimensional signature for the VCGK kernel and the h-layer entropic representation for the ASK kernel have a good ability of distinguishing vertices of graphs of different classes. On the other hand, comparing to the original shapes of the graphs, the vertex clustering results for the VCGK kernel is better distributed than those for the ASK kernel. In other words, the vertex distributions for the VCGK kernel are more like fishes. This indicates that the proposed VCGK kernel can better preserve the vertex information residing on the graph structures than the ASK kernel. In terms of the vertex alignment results, Fig.4-b, Fig.4-c, and Fig.4-d indicate that the number of aligned vertex pairs identified by the VCGK kernel for graphs from the same class is much more than that for the graphs from different classes. By contrast, Fig.5-b, Fig.5-c, and Fig.5-d indicate that the number of aligned vertex pairs identified by the ASK kernel are quite similar for graphs from either the same class or the different classes. These observations indicate that the proposed kernel VCGK has better ability of distinguishing graphs from different classes than the ASK kernel. Moreover, we observe that the VCGK kernel can identify more pairs of vertices for graphs from the same class than the ASK. This indicates that the proposed VCGK kernel has better ability of enhancing the similarity measure than the ASK kernel. Finally, we observe that the aligned vertices identified by the VCGK kernel between graphs of different classes are distributed in the small local region. By contrast, such aligned vertices identified by the ASK kernel are distributed over the global region. As a result, the VCGK kernel can better express the differences between graphs over global structures. This again indicates that the VCGK kernel can better identify graphs of different classes.

B. Graph Classification

We evaluate the performance of the vertex clustering graph kernel (VCGK) on graph classification problems. Furthermore, we also compare our kernel with three state-of-the-art kernels, including 1) the Weisfeiler-Lehman subtree kernel (WLSK) [16], 2) the shortest path graph kernel (SPGK) [13], and 3) the aligned subtree kernel (ASK). Both the WLSK and ASK kernels require a tree-index method for strengthening the vertex label. In this paper, we set the highest dimension h for their tree-index methods (i.e., the height of the subtrees identified by the tree-index methods) as 10. Since the classification accuracies of the WLSK and ASK kernels tend to be stable when the highest dimension h is greater than 8. We compute the kernel matrix associated with each kernel on each dataset. We perform 10-fold cross-validation using the C-Support Vector Machine (C-SVM) Classification to compute the classification accuracies, using LIBSVM [26]. We use nine samples for training and one for testing. All the C-SVMs were performed along with their parameters optimized on each dataset. We report the average classification accuracy ($\pm$ standard error) and the runtime for each kernel in Table I. The runtime is measured under Matlab R2011a running on a 2.5GHz Intel 2-Core processor (i.e., i5-3210m).

In terms of classification accuracies, our kernel outperforms all the alternative kernels, only the ASK kernel is a little higher than our kernel on the BAR31 dataset. Especially, the classification accuracy of our VCGK kernel is much higher than that of other kernels on the challenging GatorBait database. Note that, as we have stated, the GatorBait dataset has only 100 samples, but is divided as 30 classes. Thus this dataset is extremely hard for classification. These observations indicate that the VCGK kernel has a good ability of distinguishing graphs from different classes. The reasons for this effectiveness are threefold. First, unlike the WLSK and SPGK kernels that ignore correspondence information between substructures, the VCGK kernel establishes reliable structural correspondences between substructures and vertices, i.e., our kernel can identify the structural correspondence information based visual background on computer vision databases. Second, compared to the ASK kernel that also establishes correspondence information, only our VCGK kernel can guarantee the transitivity between pairs of aligned substructures/vertices. Third, unlike all alternative kernels that only reflect characteristics for each pair of graphs, our kernel encapsulates the information from all graphs. i.e., our kernel can be seen as an instance of transductive learning [27], where all graphs (including the training and test sets) are used to compute the kernel matrix. However, note that we do not observe the class labels of the test graphs during the training. As a result, our kernel can encapsulate richer graph characteristics than other kernels. In terms of the runtime, our kernel can finish the computation in a polynomial time, though it is not the fastest kernel.

V. Conclusions

In this paper, we have proposed a new graph kernel based on vertex clustering. We theoretically show that this kernel not only overcomes the shortcoming of ignoring correspondence information between isomorphic substructures that arises in most existing graph kernels, but also guarantees the transitivity between the correspondence information. Experiments

Preprint submitted to 23rd International Conference on Pattern Recognition.
Received April 17, 2016.

TABLE I. CLASSIFICATION ACCURACIES (IN % ± STANDARD ERROR) AND THE RUNTIME OF COMPUTING KERNEL MATRICES.

Datasets	VCGK	WLSK	SPGK	ASK
BAR31	69.30 ± .52	58.53 ± .53	55.73 ± .44	**73.10** ± .67
BSP31	**63.56** ± .60	42.10 ± .68	48.20 ± .76	60.30 ± .44
GEOD31	**48.93** ± .41	38.20 ± .68	38.40 ± .65	46.21 ± .69
GatorBait	**18.60** ± .77	10.10 ± .61	9.00 ± .75	8.40 ± .83
Shock	**51.00** ± .87	36.40 ± 1.0	37.88 ± .93	42.14 ± .73

Datasets	VCGK	WLSK	SPGK	ASK
BAR31	10′20”	30”	11”	8′40”
BSP31	11′32”	25”	14”	12′40”
GEOD31	7′5”	15”	11”	14′50”
GatorBait	5′29”	33”	2′25”	8′7”
Shock	3”	3”	1”	9”

demonstrate the effectiveness of our kernel.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (Grant no.61503422), the Open Projects Program of National Laboratory of Pattern Recognition, and the program for innovation research in Central University of Finance and Economics. Edwin R. Hancock is supported by a Royal Society Wolfson Research Merit Award.

REFERENCES

[1] V. Barra and S. Biasotti, “3d shape retrieval and classification using multiple kernel learning on extended reeb graphs,” *The Visual Computer*, vol. 30, no. 11, pp. 1247–1259, 2014.

[2] F. Galasso, M. Keuper, T. Brox, and B. Schiele, “Spectral graph reduction for efficient image and streaming video segmentation,” in *Proceedings of CVPR*, 2014, pp. 49–56.

[3] J. Yan, C. Zhang, H. Zha, W. Liu, X. Yang, and S. M. Chu, “Discrete hyper-graph matching,” in *Proceedings of CVPR*, 2015.

[4] A. Khoreva, F. Galasso, M. Hein, and B. Schiele, “Classifier based graph construction for video segmentation,” in *Proceedings of CVPR*, 2015, pp. 951–960.

[5] Z. Harchaoui and F. R. Bach, “Image classification with segmentation graph kernels,” in *Proceedings of CVPR*, 2007.

[6] T. Gärtner, P. Flach, and S. Wrobel, “On graph kernels: Hardness results and efficient alternatives,” in *Proceedings of COLT*, 2003, pp. 129–143.

[7] M. Urry and P. Sollich, “Random walk kernels and learning curves for gaussian process regression on random graphs,” *Journal of Machine Learning Research*, vol. 14, no. 1, pp. 1801–1835, 2013.

[8] M. A. Alvarez, X. Qi, and C. Yan, “A shortest-path graph kernel for estimating gene product semantic similarity,” *J. Biomedical Semantics*, vol. 2, p. 3, 2011.

[9] F. Costa and K. D. Grave, “Fast neighborhood subgraph pairwise distance kernel,” in *Proceedings of ICML*, 2010, pp. 255–262.

[10] A. Gaidon, Z. Harchaoui, and C. Schmid, “A time series kernel for action recognition,” in *Proceedings of BMVC*, 2011, pp. 1–11.

[11] F. R. Bach, “Graph kernels between point clouds,” in *Proceedings of ICML*, 2008, pp. 25–32.

[12] L. Wang and H. Sahbi, “Directed acyclic graph kernels for action recognition,” in *Proceedings of ICCV*, 2013, pp. 3168–3175.

[13] K. M. Borgwardt and H. Kriegel, “Shortest-path kernels on graphs,” in *Proceedings of ICDM*, 2005, pp. 74–81.

[14] F. Aziz, R. C. Wilson, and E. R. Hancock, “Backtrackless walks on a graph,” *IEEE Trans. Neural Netw. Learning Syst.*, vol. 24, no. 6, pp. 977–989, 2013.

[15] F. Johansson, V. Jethava, D. P. Dubhashi, and C. Bhattacharyya, “Global graph kernels using geometric embeddings,” in *Proceedings of ICML*, 2014, pp. 694–702.

[16] N. Shervashidze, P. Schweitzer, E. J. van Leeuwen, K. Mehlhorn, and K. M. Borgwardt, “Weisfeiler-lehman graph kernels,” *Journal of Machine Learning Research*, vol. 12, pp. 2539–2561, 2011.

[17] N. Kriege and P. Mutzel, “Subgraph matching kernels for attributed graphs,” in *Proceedings of the 29th International Conference on Machine Learning, ICML 2012, Edinburgh, Scotland, UK, June 26 - July 1, 2012*, 2012. [Online]. Available: http://icml.cc/discuss/2012/542.html

[18] A. Feragen, N. Kasenburg, J. Petersen, M. de Bruijne, and K. M. Borgwardt, “Scalable kernels for graphs with continuous attributes,” in *Proceedings of NIPS*, 2013, pp. 216–224.

[19] L. Bai, L. Rossi, Z. Zhang, and E. R. Hancock, “An aligned subtree kernel for weighted graphs,” in *Proceedings of ICML*, 2015, pp. 30–39.

[20] H. Fröhlich, J. K. Wegner, F. Sieker, and A. Zell, “Optimal assignment kernels for attributed molecular graphs,” in *Proceedings of ICML*, 2005, pp. 225–232.

[21] I. Witten, E. Frank, and M. Hall, “Data mining: Practical machine learning tools and techniques,” 2011.

[22] D. Cai, “The fastest matlab implementation of the k-means,” in *Software available at http://www.zjucadcg.cn/dengcai/Data/Clustering.html*, 2012.

[23] S. Biasotti, S. Marini, M. Mortara, G. Patanè, M. Spagnuolo, and B. Falcidieno, “3d shape matching through topological structures,” in *Proceedings of DGCI*, 2003, pp. 194–203.

[24] P. J. Giblin and B. B. Kimia, “On the local form and transitions of symmetry sets, medial axes, and shocks,” *International Journal of Computer Vision*, vol. 54, no. 1-3, pp. 143–157, 2003.

[25] A. M. Andrew, “Independent component analysis: Principles and practice,” *Robotica*, vol. 19, no. 4, pp. 459–462, 2001.

[26] C.-C. Chang and C.-J. Lin, “Libsvm: A library for support vector machines,” *Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm*, 2011.

[27] A. Gammerman, V. Vovk, and V. Vapnik, “Learning by transduction,” in *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., 1998, pp. 148–155.