Context-Dependent Random Walk Graph Kernels and Tree Pattern Graph Matching Kernels with Applications to Action Recognition

Weiming Hu, Baoxin Wu, Pei Wang, and Chunfeng Yuan
(CAS Center for Excellence in Brain Science and Intelligence Technology, National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences; University of Chinese Academy of Sciences, Beijing 100190)
{wmhu, bxwu, pei.wang, cfyuan }@nlpr.ia.ac.cn

Yangxi Li
(National Computer network Emergency Response technical Team/Coordination Center of China, Beijing 100055)
liyangxi@outlook.com

Stephen Maybank
(Department of Computer Science and Information Systems, Birkbeck College, Malet Street, London WC1E 7HX)
sjmaybank@dcs.bbk.ac.uk

Abstract: Graphs are effective tools for modeling complex data. Setting out from two basic substructures, random walks and trees, we propose a new family of context-dependent random walk graph kernels and a new family of tree pattern graph matching kernels. In our context-dependent graph kernels, context information is incorporated into primary random walk groups. A multiple kernel learning algorithm with a proposed $l_{1,2}$-norm regularization is applied to combine context-dependent graph kernels of different orders. This improves the similarity measurement between graphs. In our tree-pattern graph matching kernel, a quadratic optimization with a sparse constraint is proposed to select the correctly matched tree-pattern groups. This augments the discriminative power of the tree-pattern graph matching. We apply the proposed kernels to human action recognition, where each action is represented by two graphs which record the spatiotemporal relations between local feature vectors. Experimental comparisons with state-of-the-art algorithms on several benchmark datasets demonstrate the effectiveness of the proposed kernels for recognizing human actions. It is shown that our kernel based on tree-pattern groups, which have more complex structures and exploit more local topologies of graphs than random walks, yields more accurate results but requires more runtime than the context-dependent walk graph kernel.

Index terms: Visual action recognition, Graph kernel, Graph matching, Contexts, Tree patterns

1. Introduction

Human action recognition [58, 59] is one of the most challenging issues in computer vision. It has very wide applications in domains such as visual surveillance, video retrieval, human-computer interaction, and medical monitoring. Many methods for human action recognition represent each action as an ensemble of local spatiotemporal feature vectors [4, 11, 17, 22] corresponding to sparse interest points extracted from videos, and carry out human action recognition by analyzing these spatiotemporal feature vectors. Bag-of-visual words (BoW)-based [4, 22] methods and methods based on the vector of local aggregated descriptors (VLAD) [51] are popularly used to statistically represent an ensemble of local feature vectors as a single vector. These vectors are used to construct an action classifier. However, the BoW and VLAD-based methods do not make use of the spatiotemporal relations between local feature vectors. The subdivision of videos into grids [52] or statistical measures of the concurrency [53] of local feature vectors were used to

include information about spatiotemporal relations in the BoW or VLAD-based methods. However, the complex structure of the local feature vectors is still not fully exploited.

Graphs are effective tools for structuring and modeling complex data [1, 7]. The vertices represent data themselves and the edges represent their relations. It is interesting to investigate human action recognition using graphs for modeling ensembles of local feature vectors. To this end, there are two nontrivial difficulties to be solved: how to construct graphs to model these local feature vectors and how to measure the similarity between the constructed graphs. In this paper, we focus on these two problems and propose new graph-based methods for human action recognition.

1.1. Related work

We briefly review graph-based action recognition and graph similarity measurement.

1.1.1. Graph similarity measurement

The traditional statistics-based classification methods cannot be directly used to classify actions represented by structured graphs. Graph similarity measurement bridges the gap between structured action representation and statistical classification. Graph kernels and graph matching are the main techniques for measuring similarities between graphs.

Graph kernel-based measures of the similarity between two graphs usually compare all pairs of substructures of the graphs. According to the form of substructures, traditional graph kernels can be categorized as random walks-based [5, 20, 55], trees-based [26, 56], and limited-size sub-graphs- based [57]. Random walk-based graph kernels have received increasing attention recently. Gartner et al. [5] computed the graph kernel of two labeled graphs by counting the number of matched labeled random walks. The method was extended by Borgwardt et al. [1] by replacing the Dirac kernel with more complex kernels for continuous attributes. Vishwanathan et al. [20] proposed several techniques to speed up the computation of random walk graph kernels. Harchaoui and Bach [8] built a set of segmentation graph kernels on images and utilized a multiple kernel learning method to combine these kernels for classifying images. Trees-based graph kernels [26, 56] decompose graphs into trees as substructures. Mahe and Vert [26] measured the similarity between graphs by counting the number of tree substructures with the same labels in the graphs. Shervashidze et al. [56] defined an efficient kernel by comparing sub-tree-like patterns. Limited-size sub-graphs-based kernels [57] decompose the two graphs into a series of substructures with specific sizes. Shervashidze et al. [57] divided each graph into a number of graphlets and compared the numbers of the graphlets with the same structure in order to measure graph similarities. All of the above graph kernels are built by comparing the similarities between all pairs of substructures, such as walks, from the two graphs. The contexts of the substructures are not exploited. Moreover, they can all be viewed as summation kernels on substructures [34] that do not take into consideration the correspondences between substructures.

Graph matching [28, 29, 30, 31] determines a mapping between vertices of two graphs such that the structure of the relations between vertices is preserved as much as possible. Leordeanu and Hebert [31] proposed a spectral method to solve the graph matching problem. Egozi et al. [30] presented a probabilistic interpretation for graph matching and derived a probabilistic graph matching algorithm. Cho et al. [28] proposed a max-pooling strategy for graph matching. However, these algorithms do not consider high order topological information which is useful for improving the performance of the matching

1.1.2. Graph-based action recognition

Graphs have been applied to model ensembles of local feature vectors or patches for human action recognition. Borzeshi et al. [2] represented each frame as a graph whose vertices correspond to the spatial local feature vectors extracted from the frame. Raja et al. [14] described a person in a frame using a graphical model whose vertices encode the positions of human body parts and the action label. Gaur et al. [6] constructed a string of feature graphs for representing the spatiotemporal layout of local feature vectors, where each graph models the spatial configuration of local feature vectors in a small temporal segment. Ma et al. [40] proposed an excellent method for action recognition by finding a compact set of hierarchical space-time tree structures of human actions from videos. Ta et al. [18] constructed a hypergraph to model the extracted spatiotemporal local feature vectors in a video. A hypergraph matching algorithm was used for action recognition. The above methods construct graphs to model local feature vectors or body parts. They do not explicitly model the spatiotemporal relations between these local features or body parts. A number of researchers represent a video action using a graph, and then recognize human actions using graph comparison. Celiktutan et al. [3] found vertex correspondences between graphs through graph matching. Then, the similarity between two graphs was computed by summing the similarities between all the correctly matched vertices. However, the graph similarity measurement is based on only a quite small number of vertices, and it cannot completely characterize the graphs. Wang et al. [36] and Aoun et al [54] constructed graph kernels for similarity measurements between graphs for action recognition. However, these graph kernels are based on random walks and their contexts are not utilized. Moreover, the correspondences between the substructures were not considered.

1.2. Our work

With the aim of handling the limitations in previous graph-based action recognition methods as well as the limitations in previous graph similarity measurement methods, we propose a context-dependent random walk graph kernel [37] and a tree pattern matching kernel for human action recognition.

We construct two directed attributed graphs, the concurrent graph and the causal graph, to describe the spatiotemporal layouts of the local feature vectors extracted from each action video. The vertex attributes in both graphs are the local feature vectors. The edge attributes in the concurrent and causal graphs describe the relations of the local feature vectors within a frame and between frames respectively.

Setting out from two basic substructures, random walks and trees, we propose a context-dependent random walk graph kernel and a tree pattern matching kernel. While these two new kernels decompose graphs into primary random walk groups and tree-pattern groups respectively, they are constructed by combining the vertex kernels and edge kernels in a similar way. In the context-dependent random walk graph kernel, we propose to use a direct product for computing the context-dependent similarities between primary walk groups. We utilize a generalized multiple kernel learning algorithm with the $l_{1,2}$-norm regularization to determine the weights for combining context-dependent graph kernels of different orders. The proposed $l_{1,2}$-norm regularization imposes the sparseness constraint on different orders' context-dependent kernels from the same graphs (concurrent graphs or causal graphs), because these graph kernels may contain redundant information, in particular when the order is high. The regularization adds the smoothness constraint on kernels from the concurrent graphs and the causal graphs respectively, to ensure that both the concurrent graphs and the causal

graphs which preserve different relations between vertices are used to estimate the similarity between the videos. In our tree pattern graph matching kernel, the similarity between tree pattern groups is computed recursively in a dynamic programming formulation. We formulate the correspondences between tree pattern groups as a quadratic optimization problem with a sparse constraint. Only the correctly matched tree-pattern groups are incorporated into graph matching. We apply the context-dependent random walk graph kernel and the tree pattern graph matching kernel to measure the similarity between human actions for action recognition. A SVM-based classifier is learnt for action recognition. As trees have more complex structures than random walk, the tree pattern graph matching kernel yields more accurate results, but requires more runtime, than the context-dependent random walk graph kernel.

The main novelties of our work are summarized as follows:

- The proposed two graphs for representing human actions are complementary to each other. Compared with the popular BoW-based models [4, 22], they not only preserve the individual properties of local feature vectors but also capture the spatiotemporal relations among them, and hence effectively provide a more informative representation for actions.
- Compared with traditional random walk kernels which use the same weight to combine all the pairs of primary walk groups [1, 5, 20], the proposed context-dependent random walk graph kernel weights the pairs of primary walk groups using their contexts and then improves the similarity measurement between graphs. The generalized multiple kernel learning algorithm effectively selects and combines informative context- dependent graph kernels.
- The proposed tree-pattern groups preserve more local structural information in the graphs. The proposed tree pattern graph matching kernel suppresses errors caused by falsely matched tree-pattern groups and increases the discriminative power of the kernel.

The rest of the paper is organized as follows: Section 2 proposes our context-dependent random walk graph kernel. Section 3 presents our tree pattern graph matching kernel. Section 4 describes the concurrent and causal graphs-based action recognition methods. Section 5 reports the experimental results. Section 6 concludes the paper.

2. Context-Dependent Random Walk Graph Kernel

We first describe how to define context-dependent random walk graph kernels of different orders, then show how to compute these kernels using the direct product graph, and finally show how to learn the weights of these kernels using the generalized multiple kernel learning.

2.1. Definition

We define context-dependent random walk graph kernels on the basis of primary random walk groups and their contexts in directed attributed graphs.

2.1.1. Primary walk groups on directed attributed graphs

In a directed attributed graph $G=(V,E)$ with the set of N vertices $\{v_i\}_{i=1}^{N}$ and the edge set E, a vertex v_i is a point with a coordinate vector in a Euclidean space and an attribute vector in the feature space. An ordered pair of vertices v_i and v_j defines an edge $(v_i,v_j)\in E$. Each edge is associated with an attribute vector.

A random walk *w* with length *n* from graph *G* is defined as a sequence of vertices connected by *n* edges: $w=(v_{w_0},e_{w_1},v_{w_1},...,e_{w_n},v_{w_n})$, where $e_{w_i}\ (1\le i\le n)$ is the edge connecting vertices $v_{w_{i-1}}$ and v_{w_i}. A primary walk group $\rho_G^n(i,j)$ with length *n* in graph *G* is defined as the set of random walks with length *n* starting at vertex v_i and ending at v_j.

Let $k_v(v,v')$ be the basic kernel function measuring the similarity between vertices *v* and v' in graphs *G* and G' respectively. Let $k_e(e,e')$ be the basic kernel function measuring the similarity between edges *e* and e' in graphs *G* and G' respectively. These two basic functions are designed using the coordinate vectors and the attribute vectors of vertices and the attribute vectors of edges according to the task at hand (See Section 4.2). A kernel function $k_w(w,w')$ which measures the similarity between any two walks *w* and w' with the same length *n* is defined in the following way. If the length *n* is 0, then walks *w* and w' are vertices v_{w_0} and $v'_{w'}$ and the kernel $k_w(w,w')$ equals to the vertex kernel $k_v(v_w,v'_{w'})$. If length *n* is larger than 1, then $k_w(w,w')$ is defined as the product of the kernels of the vertices and the kernels of the edges along the two walks *w* and w' respectively:

$$k_w(w,w')=\prod_{i=0}^{n}k_v(v_{w_i},v'_{w'_i})\prod_{j=1}^{n}k_e(e_{w_j},e'_{w'_j}) \tag{1}$$

We define a kernel k_{wg} for measuring the similarity between any two primary walk groups $\rho_G^n(i,j)$ and $\rho_{G'}^n(r,s)$ with the same length *n* as a summation of kernels of all the pairs of walks from these two primary walk groups:

$$k_{wg}(\rho_G^n(i,j),\rho_{G'}^n(r,s))=\sum_{w\in\rho_G^n(i,j)}\sum_{w'\in\rho_{G'}^n(r,s)}k_w(w,w')\,. \tag{2}$$

2.1.2. Contexts of primary walk groups

We define the contexts of a primary walk group, based on the contexts of vertices. The context $c(i)$ of vertex v_i is defined as the set of a fixed number of vertices which are nearest to v_i in the Euclidean coordinate space. Then, the context $\eta_G^n(i,j)$ of a primary walk group $\rho_G^n(i,j)$ consists of primary walk groups starting at the contexts of v_i and ending at the context of v_j.

We define the kernel for the contexts $\eta_G^n(i,j)$ and $\eta_{G'}^n(r,s)$ of primary walk groups $\rho_G^n(i,j)$ and $\rho_{G'}^n(r,s)$ as the sum of the kernels between the primary walk groups in these two primary walk group contexts respectively:

$$k_{wg}\left(\eta_G^n(i,j),\eta_{G'}^n(r,\mathrm{s})\right)=\sum_{\substack{f\in c(i),g\in c(j),\\ o\in c(r),q\in c(s)}}k_{wg}\left(\rho_G^n(f,g),\rho_{G'}^n(o,q)\right). \tag{3}$$

As similar primary walk groups usually have similar contexts, we weight the kernel for primary walk groups using the kernel for their contexts. The context-dependent kernel k_{cwg} for primary walk groups $\rho_G^n(i,j)$ and $\rho_{G'}^n(r,s)$ is defined as the sum of the kernel between $\rho_G^n(i,j)$ and $\rho_{G'}^n(r,s)$ and the kernel weighted by their contexts:

$$k_{cwg}\left(\rho_G^n(i,j),\rho_{G'}^n(r,s)\right)=k_{wg}\left(\rho_G^n(i,j),\rho_{G'}^n(r,s)\right)\left(1+\kappa k_{wg}\left(\eta_G^n(i,j),\eta_{G'}^n(r,s)\right)\right) \tag{4}$$

where the "1" is used to keep the similarity between the primary walk groups themselves and the parameter $\kappa>0$ controls the degree of the effect of the context information to the context-dependent kernel. The more the context similarity, the more the context-dependent kernel is increased. The parameters κ is determined by cross-validation. Fig. 1 shows an example of the contexts of primary walk groups, the corresponding kernels, and the context dependent kernel.

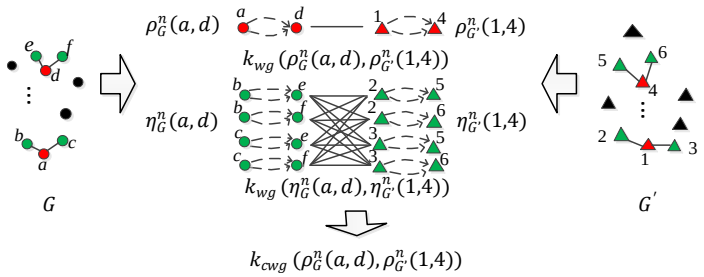


Fig. 1. An example of the contexts of primary walk groups and the corresponding kernels: The left and right columns show two graphs G and G', where the green vertices are the contexts of the red ones (The edges are not shown); The middle column shows the kernels on primary walk groups and the kernels on their contexts, both of which are used to define the context-dependent kernel on primary walk groups. A directed dashed curve denotes a random walk.

2.1.3. Context-dependent random walk graph kernel

The *n*th-order context dependent walk graph kernel is defined as the mean of the sum of the context dependent walk graph kernels with walk length *n*:

$$k_g^n(G,G')=\frac{1}{N_G^n N_{G'}^n}\sum_{\substack{\rho_G^n(i,j)\subset\rho_G^n\\ \rho_{G'}^n(r,s)\subset\rho_{G'}^n}} k_{cwg}\left(\rho_G^n(i,j),\rho_{G'}^n(r,s)\right), \tag{5}$$

where N_G^n is the number of primary walk groups with length *n* in graph *G*, and ρ_G^n is the set of all the walks with length *n* in graph *G*. The normalization by $N_G^n N_{G'}^n$ takes account of the fact that there are different numbers of local feature vectors in different graphs.

We use a positive weight λ^n to emphasize the importance of the *n*th-order context dependent random walk graph kernel. Then, the final graph kernel is defined as:

$$k_g(G,G')=\sum_n \lambda^n k_g^n(G,G'). \tag{6}$$

Our context-dependent random walk graph kernel is relevant to the traditional random walk graph kernel and the context-dependent kernel for attributed point sets [13]. The description of their relations is included in Appendix A, which is available online.

2.2. Direct product graph-based computation

In practice, a direct product graph [9] is used to efficiently compute the random walk kernel between two graphs. Correspondingly, we propose to utilize direct product graphs to compute context-dependent random walk graph kernels of different orders.

2.2.1. Direct product graph

The direct product graph $G_d=(V_d,E_d)$ of two graphs $G=(V,E)$ and $G'=(V',E')$ is a graph whose vertices are in the set of pairs of the vertices in *G* and G' respectively, as shown in Fig. 2. For a vertex v_i

in G and a vertex v'_r in G', if the vertex kernel $k_v(v_i, v'_r)$ between these two vertices is larger than 0, then the vertex pair (v_i, v'_r) forms a vertex in G_d. For an edge (v_i, v_j) connecting vertices v_i and v_j in G and an edge (v'_r, v'_s) in G', if the edge kernel $k_e((v_i, v_j), (v'_r, v'_s))$ between these two edges is larger than 0, then there is an edge connecting vertices (v_i, v'_r) and (v_j, v'_s) in G_d. Each vertex (v_i, v'_r) in G_d is assigned a weight w_{ir} which equals $k_v(v_i, v'_r)$. Each edge $((v_i, v'_r), (v_j, v'_s))$ in G_d is assigned a weight $w_{ir,js}$ which equals $k_e((v_i, v_j), (v'_r, v'_s))$. For each edge in G_d, there exists a corresponding edge in G and a corresponding edge in G'. For each random walk in G_d, there exists a corresponding random walk in G and a corresponding random walk in G', both with the same length.

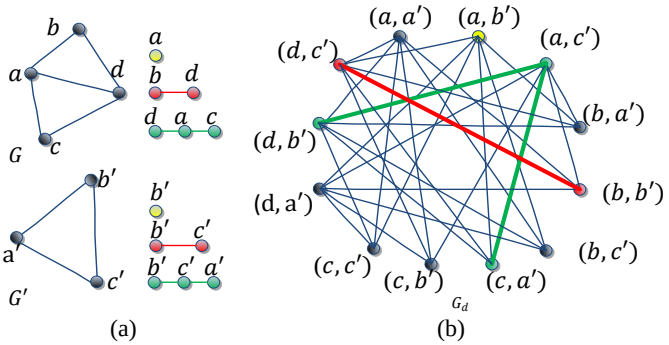


Fig. 2. Direct product graph: (a) Two graphs G and G' with random walks with different lengths; (b) The corresponding direct product graph and random walks: the yellow, red, and green colors represent, respectively, a 0-order random walk, a 1-order random walk, and a 2-order random walk.

For the direct product graph G_d, we construct a diagonal matrix $\mathbf{W}_{V_d} \in \mathbb{R}^{|V_d|\times|V_d|}$, in which the ir-th diagonal element $W_{V_d}(ir, ir)$ is w_{ir}, to contain the vertex weights. We construct a matrix $\mathbf{W}_{E_d} \in \mathbb{R}^{|V_d|\times|V_d|}$, in which the element $W_{E_d}(ir, js)$ is $w_{ir,js}$, to contain the edge weights. The nth-order weight matrix $\mathbf{W}_d^n$ of G_d is defined as $\mathbf{W}_d^n = \mathbf{W}_{V_d}(\mathbf{W}_{E_d}\mathbf{W}_{V_d})^n$. According to (1), $\mathbf{W}_{E_d}\mathbf{W}_{V_d}$ describes the changes in the kernels of random walks when the walks are extended one edge. Simple derivation yields $W_d^n(ir, js) = k_{wg}(\rho_G^n(i, j), \rho_{G'}^n(r, s))$ (For details, see Appendix B, which is available online). Therefore, each nonzero element in $\mathbf{W}_d^n$ is the similarity between the corresponding primary walk groups with length n in graphs G and G', respectively.

2.2.2. Computation of context-dependent graph kernels of different orders

To represent the context of each vertex in G_d, we define the context matrix $\mathbf{C}_d \in \mathbb{R}^{|V_d|\times|V_d|}$ whose elements $C_d(ir, js)$ are 1 if v_j is a context of v_i in G and v'_s is a context of v_r in G', otherwise are 0. Each row in $\mathbf{C}_d$ describes the context of one vertex. The nth-order context-dependent weight matrix $\mathbf{W}_{cd}^n$ of G_d is computed by $\mathbf{W}_{cd}^n = \mathbf{W}_d^n + \kappa\mathbf{W}_d^n \odot (\mathbf{C}_d\mathbf{W}_d^n\mathbf{C}_d^T)$, where $\odot$ is the Hadamard product.

According to (4), each nonzero element in $\mathbf{W}_{cd}^n$ corresponds to the context-dependent kernel between two primary walk groups with length n in graphs G and G' respectively, i.e., $W_{cd}^n(ir, js) = k_{cwg}(\rho_G^n(i, j), \rho_{G'}^n(r, s))$, where k_{cwg} is defined in (4) (For details, see Appendix B). According to (5), the nth-order context-dependent

walk graph kernel on graphs G and G' is rewritten as:

$$k_g^n(G,G') = \frac{1}{N_G^n N_{G'}^n} \sum_{ir,js} W_{cd}^n(ir,js) . \tag{7}$$

It is apparent that $N_G^n N_{G'}^n$ equals to the number of the nonzero entries in $\mathbf{W}_{cd}^n$. The computational complexity of the context-dependent walk graph kernel can be found in Appendix C.

2.3. Generalized multiple kernel learning

Substitution of (7) into (6) yields the final graph kernel between two graphs. The weights $\{\lambda^n\}$ in (6) can be estimated using the labeled samples. We apply the generalized multiple kernel learning [19] to this estimation. The multiple kernel learning is an information fusion method. Each type of information corresponds to one kernel. We use the multi-kernel learning to combine the graph kernels of different orders.

In real applications, each sample can be represented by a set of L graphs $\{G_l\}_{l=1}^L$ which have the same vertex set, where different graphs represent different characteristics of these vertices (In our action recognition method, L equals 2, i.e., a concurrent graph and a causal graph are used to represent a sample, as described in Section 4.1). It is apparent that the L graphs share the same 0th order kernel k_g^0. Let Z_l be the maximal order of the context-dependent walk group kernels on graphs l. The kernel on any two samples S and S' is defined as:

$$k(S,S') = \lambda^0 k_g^0(G_1,G_1') + \sum_{l=1}^{L} \sum_{z=1}^{Z_l} \lambda_l^z k_g^z(G_l,G_l') , \tag{8}$$

where the $\{k_g^z(G_l,G_l')\}_{z=0}^{Z_l}$ are computed by (7).

It is assumed that a set of M training samples $\{S_m\}_{m=1}^M$ is available with labels $\{y_m\}_{m=1}^M$. We define a set of $M \times M$ kernel matrices $\{\mathbf{K}^0, \{\mathbf{K}_l^z\}_{z=1}^{Z_l}\}_{l=1}^L$ for the training samples, where $K^0(S_i,S_j) = k_g^0(G_{l,S_i}, G_{i,S_j})$ and $K_l^z(S_i,S_j) = k_g^z(G_{l,S_i}, G_{l,S_j})$. Corresponding to (8), we define the kernel matrix $\mathbf{K}$ for the training samples as follows:

$$\mathbf{K} = \lambda^0 \mathbf{K}^0 + \sum_{l=1}^{L} \sum_{z=1}^{Z_l} \lambda_l^z \mathbf{K}_l^z , \tag{9}$$

where $\lambda_l^z > 0$ is the weight for $\mathbf{K}_l^z$. Let $\mathbf{\Lambda}$ be the weight vector whose elements are $\{\lambda^0, \{\lambda_l^z\}_{l=1,\ldots,L}^{z=1,\ldots,Z_l}\}$. Let $\mathbf{Y}$ be an $M \times M$ diagonal matrix with the numbers indicating the sample labels $\{y_m\}_{m=1}^M$ on the diagonal. Let $\mathbf{1}$ be an M-dimensional vector in which all the entries are 1. The dual problem of generalized multi-kernel learning [19] is represented as minimizing the objective function $D(\mathbf{\Lambda})$ which is defined as:

$$\begin{aligned} &D(\mathbf{\Lambda}) = \max_{\boldsymbol{\alpha}} \left(\mathbf{1}^T \boldsymbol{\alpha} - \tfrac{1}{2} \boldsymbol{\alpha}^T \mathbf{Y}\mathbf{K}\mathbf{Y}\boldsymbol{\alpha} \right) \\ &\text{Subject to } \mathbf{1}^T \mathbf{Y}\boldsymbol{\alpha} = 0, 0 \le \alpha_m \le C_1, \end{aligned} \tag{10}$$

where $\boldsymbol{\alpha} = (\alpha_m)_{m=1}^M$ is the Lagrangian multiplier vector, C_1 is a constant controlling the importance of the loss, and the weights $\{\lambda\}$ are included in $\mathbf{K}$ and $\mathbf{\Lambda}$.

We add a regularization on the weights $\{\lambda^n\}$. On the one hand, context-dependent graph kernels of different orders on the same graphs (the lth graphs) may contain redundant information for classifying

samples. So, we add a sparseness constraint on the weights of those kernels. On the other hand, different graphs preserve different relations between vertices, and contain complementary information. So, we add a smoothness constraint on the weights of the kernels from different graphs (the L graphs for each sample), as well as the weight for the 0-th order context dependent walk graph kernel. Therefore, we propose a $l_{1,2}$-norm regularization on the kernel weights $\mathbf{\Lambda}$. It is defined as:

$$r(\mathbf{\Lambda}) = \frac{1}{2}\left\| \left| \lambda^0 \right|_1, \left| (\lambda_1^1, ..., \lambda_1^{Z_1}) \right|_1, ..., \left| (\lambda_l^1, ..., \lambda_l^{Z_l}) \right|_1, ..., \left| (\lambda_L^1, ..., \lambda_L^{Z_L}) \right|_1 \right\|_2^2. \tag{11}$$

We add the $l_{1,2}$-norm regularization to the generalized multiple kernel learning framework. The objective function $D(\mathbf{\Lambda})$ is updated by:

$$D(\mathbf{\Lambda}) = \max_{\boldsymbol{\alpha}} \left(\mathbf{1}^T \boldsymbol{\alpha} - \tfrac{1}{2} \boldsymbol{\alpha}^T \mathbf{YKY} \boldsymbol{\alpha} + C_2 r(\mathbf{\Lambda}) \right), \tag{12}$$

where C_2 is the constant controlling the regularization on kernel weights.

We utilize the mini-max optimization algorithm [19] to calculate $\mathbf{\Lambda}$. The details of the algorithm can be found in Appendix D. After the weights in $\mathbf{\Lambda}$ are obtained, the similarities between samples are computed using (8). These similarities are used to train a SVM-based classifier. SVMs are appropriate for graph kernel-based recognition, because SVMs only need to input similarities between samples. The extraction of feature vectors from samples is avoided. Furthermore, SVMs allow for parallel computation of the similarities between samples. Parallel computation is popularly used for action recognition.

3. Tree-Pattern Graph Matching Kernel

Considering that random walks in a graph have simple shapes with chain structures and cannot capture sufficient topological information in a graph, we propose a graph matching kernel based on decomposing graphs into tree patterns which have more complex structure than random walks. The kernel is computed by comparing the incoming and outgoing tree-pattern groups from two graphs. We describe first how to define the tree patterns, then how to recursively measure the kernels for tree-pattern groups, and finally how to construct the tree pattern graph matching kernel.

3.1. Tree patterns

For a directed attributed graph $G(V, E)$, each vertex $v_i \in V$ has a set of incoming neighbors $\delta^-(v_i) = \{v_j \in V \mid (v_j, v_i) \in E\}$ and a set of outgoing neighbors $\delta^+(v_i) = \{v_j \in V \mid (v_i, v_j) \in E\}$. We define the in-degree of vertex v_i as the number of its in-coming neighbors, and define its out-degree as the number of its out-going neighbors.

A tree is a directed acyclic connected graph. It is denoted as $t = (U_t, F_t)$ where U_t is its vertex set and F_t is its edge set. The vertices with in-degree zero are called root vertices. The vertices with out-degree zero are called leaf vertices. Trees are naturally oriented by directed edges from root vertices to leaf vertices. The height $h(t)$ of a tree t is defined as the maximum number of edges connecting a leaf vertex and a root vertex plus one. The branch $b(t)$ of a tree is defined as the absolute value of difference between the number of root vertices and the number of leaf vertices. The height and the branch are used to describe the complexity of a tree.

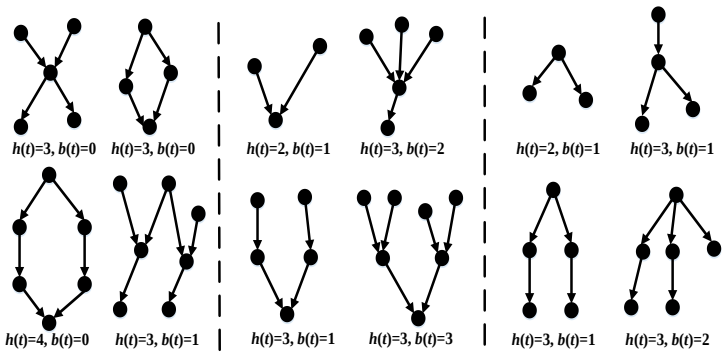
A tree pattern [26] is a derivative of a tree, which emphasizes the tree structure. A tree-pattern from a

graph is a sub-graph which has a tree structure. In a graph $G(V,E)$, a tree-pattern which has the same structure as the tree $t=(U_t,F_t)$ is denoted as $p_t=(V_t,E_t)$ with the vertex set $V_t=\{v_{t_1},v_{t_2},...,v_{t_{|U_t|}}\}$ and the set E_t of edges linking vertices in V_t, where $v_{t_i}\in V$ and $|U_t|$ is the number of vertices in tree *t*. There is a one-to-one mapping of vertices and edges between tree pattern p_t and tree *t*.

Given two tree patterns $p_t=(V_t,E_t)$ and $p'_t=(V'_t,E'_t)$ which have the same structure as the tree *t*, we measure their similarity using the similarities between the vertices and between the edges in the two tree patterns respectively. Let $v_{t_i}\in V_t$ and $v'_{t_i}\in V'_t$ correspond to the *i*th vertex of tree *t*. The tree-pattern kernel between p_t and p'_t is defined as the weighted product of vertex kernels and edge kernels:

$$k_t(p_t,p'_t)=\rho_{\mu,\gamma}(t)\prod_{i=1}^{|U_t|}k_v(v_{t_i},v'_{t_i})\prod_{j=1}^{|F_t|}k_e(e_{t_j},e'_{t_j}), \tag{13}$$

where $\rho_{\mu,\gamma}(t)=\mu^{h(t)-1}\gamma^{b(t)}$ is a weighting function which measures the structure complexity of the tree *t*. The complexity of a tree increases when its height $h(t)$ or its branch $b(t)$ increases. By adjusting μ and γ, the effect of the complexity of tree patterns on the similarity measurement can be increased or reduced.



(a) Trees (b) Incoming trees (c) Outgoing trees

Fig. 3. Examples of trees with different structures.

We consider incoming trees and outgoing trees. In an incoming tree, the out-degree of all the vertices is one except for the leaf vertex. In an outgoing tree, the in-degree of all the vertices is one except for the root vertex. Fig. 3 shows some examples of incoming and outgoing trees. The tree-patterns extracted from graphs according to incoming trees and outgoing trees are called incoming tree-patterns and outgoing tree-patterns respectively. These two kinds of tree patterns exploit, respectively, the incoming and outgoing neighborhood information on vertices. Let T_{in}^h/T_{out}^h be the set of incoming/outgoing trees whose heights are less than *h*. Let $P(t,G)$ be the set of tree patterns structured by tree *t* and extracted from graph *G*. We define the tree pattern-based *h*-order graph kernel $k_g^h(G,G')$ between graphs *G* and G' as the summation of the similarities of all the pairs of the incoming and outgoing tree patterns whose heights are less than *h*:

$$k_g^h(G,G')=\sum_{t\in T_{in}^h}\sum_{\substack{p_t\in P(t,G)\\ p'_t\in P(t,G')}}k_t(p_t,p'_t)+\sum_{t\in T_{out}^h}\sum_{\substack{p_t\in P(t,G)\\ p'_t\in P(t,G')}}k_t(p_t,p'_t). \tag{14}$$

This new kernel uses incoming and outgoing tree patterns to capture the different local neighborhood relations between vertices. In contrast with random walks, these two types of tree patterns have more complex structures and more effectively describe the local topological structure of graphs for measuring the similarities

between graphs.

3.2. Similarities between tree-pattern groups

By using each vertex as the root of outgoing trees or the leaf of incoming tress, the graph is decomposed into many tree patterns. It is an NP hard problem to extract all the tree patterns from a graph. Instead of extracting all the tree patterns, we recursively compute the similarities between graphs based on tree patterns. This computation depends on the definition of affinal incoming/outgoing tree-pattern groups and incoming/outgoing neighborhood matching sets.

Definition 1: Affinal incoming tree-pattern group: Each vertex v's h-order affinal incoming tree-pattern group $H_{in}^{h}(v)$ in a graph G is the set of incoming tree-patterns which have the same leaf vertex v and whose heights are no more than h.

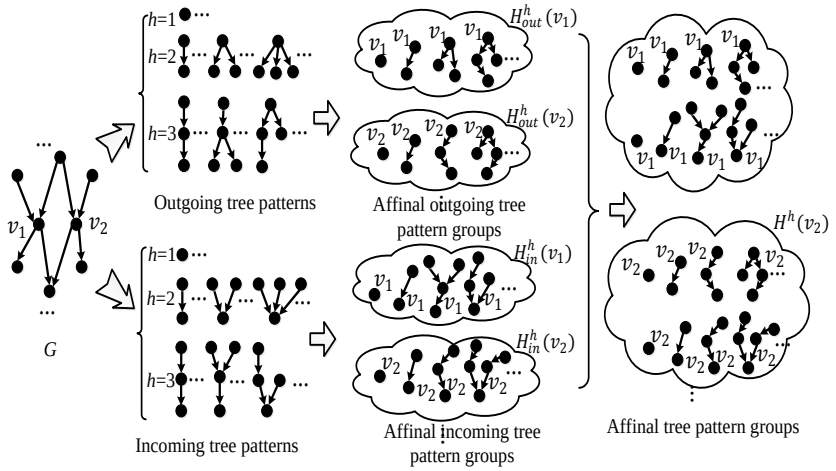
Definition 2: Affinal outgoing tree-pattern group: Each vertex v's h-order affinal outgoing tree-pattern group $H_{out}^{h}(v)$ in a graph G is the set of outgoing tree-patterns which have same root vertex v and whose heights are no more than h.

Fig. 4 shows some examples of affinal incoming/outgoing tree-pattern groups. The similarity between the affinal incoming tree-pattern groups $H_{in}^{h}(v)$ for vertex v in graph G and $H_{in}^{h}(v')$ for vertex v' in graph G' is defined as the summation of similarities between all the pairs of incoming tree-patterns taken from $H_{in}^{h}(v)$ and $H_{\text{in}}^{h}(v')$, respectively. The corresponding kernel is:

$$k_H(H_{in}^{h}(v), H_{in}^{h}(v')) = \sum_{p \in H_{in}^{h}(v)} \sum_{p' \in H_{in}^{h}(v')} k_t(p, p'), \tag{15}$$

where $k_t()$ is defined in (13) and the similarity between tree patterns with different tree structures is 0. Similarly, the kernel for affinal outgoing tree-pattern groups is defined as:

$$k_H(H_{out}^{h}(v), H_{out}^{h}(v')) = \sum_{p \in H_{out}^{h}(v)} \sum_{p' \in H_{out}^{h}(v')} k_t(p, p'). \tag{16}$$



(a) (b) (c) (d)

Fig. 4. Examples of affinal tree-patterns: (a) A directed graph G; (b) Incoming tree-patterns and outgoing tree-patterns; (c) Affinal incoming tree-pattern groups and affinal outgoing tree-pattern groups; (d) Affinal tree-pattern groups.

We estimate the h-order affinal tree-pattern group kernel between two vertices using the h-1 order affinal tree-pattern group kernels of the neighbors of the two vertices. We introduce two definitions to exploit the incoming and outgoing neighborhood information in graphs.

Definition 3: Incoming neighborhood matching set: The incoming neighborhood matching set $M^{-}(v,v')$ of two vertices *v* and v' in graphs *G* and G' respectively is a set of one-to-one matching pairs of the incoming neighbors of *v* and the incoming neighbors of v'. An element *R* of $M^{-}(v,v')$ consists of one or several pair(s) of vertices from incoming neighborhoods $\delta^{-}(v)$ and $\delta^{-}(v')$ of *v* and v' respectively. For pairs (*a*, *b*) and (*c*, *d*) in *R*, *a* is *c* if and only if *b* is *d*, i.e., there is the one-to-one matching between the incoming neighbors of *v* and v' in *R*. For each vertex pair (*a*, *b*) belonging to *R*, both the vertex kernel on the pair and the edge kernel on the pair of edges (*a*,*v*) and (b,v') have positive values, i.e., $k_v(a,b)>0$ and $k_e((a,v),(b,v'))>0$.

Definition 4: Outgoing neighborhood matching set: The outgoing neighborhood matching set $M^{+}(v,v')$ of two vertices *v* and v' in graphs *G* and G' respectively is defined by replacing the incoming neighborhood in $M^{-}(v,v')$ with the outgoing neighborhood, i.e., $M^{+}(v,v')$ is a set of one-to-one matching pairs of the outgoing neighbors of *v* and v'.

According to the definition of $M^{-}(v,v')$, by substituting (13) into (15), the affinal incoming tree-pattern group kernel $k_H(H_{in}^n(v),H_{in}^n(v'))$ is rewritten equivalently in a dynamic programming formulation [26] as:

$$k_H(H_{in}^h(v),H_{in}^h(v'))=k_v(v,v')\left(1+\mu\sum_{R\in M^{-}(v,v')}\frac{1}{\gamma}\prod_{(u,u')\in R}\gamma k_e((u,v),(u',v')k_H\left(H_{in}^{h-1}(u),H_{in}^{h-1}(u')\right)\right), \tag{17}$$

where μ and γ are the two parameters defined in (13) (See [26] for the details of the mathematical derivation of (17)). Then, the affinal incoming tree-pattern group kernel can be computed recursively. The initialization for the iteration is $k_H(H_{in}^1(v),H_{in}^1(v'))=k_v(v,v')$. Correspondingly, the affinal outgoing tree-pattern group kernel $k_H(H_{out}^n(v),H_{out}^n(v'))$ in (16) is rewritten as:

$$k_H(H_{out}^h(v),H_{out}^h(v'))=k_v(v,v')\left(1+\mu\sum_{R\in M^{+}(v,v')}\frac{1}{\gamma}\prod_{(u,u')\in R}\gamma k_e((v,u),(v',u')k_H\left(H_{out}^{h-1}(u),H_{out}^{h-1}(u')\right)\right), \tag{18}$$

where $k_H(H_{out}^1(v),H_{out}^1(v'))=k_v(v,v')$. In this way, the kernels for affinal tree-pattern groups can be computed efficiently, avoiding the exaction of all the tree patterns in graphs. Verifying whether the tree patterns of two graphs are from the same tree is carried out in the dynamic programming process.

For a vertex *v* in graph *G*, the affinal incoming tree-pattern groups $H_{in}^h(v)$ and the affinal outgoing tree-pattern groups $H_{out}^h(v)$ are collectively referred to as affinal tree-pattern groups $H^h(v)=H_{in}^h(v)\cup H_{out}^h(v)$, as shown in Fig. 4. The kernel $k_H(H^h(v),H^h(v'))$ between two affinal tree-pattern groups $H^h(v)$ and $H^h(v')$ is defined as:

$$k_H(H^h(v),H^h(v'))=k_H(H_{in}^h(v),H_{in}^h(v'))+k_H(H_{out}^h(v),H_{out}^h(v')). \tag{19}$$

The kernels for affinal tree-pattern groups are used to compute the tree pattern graph kernel in (14). For two graphs, the summation of the similarities between all the incoming/outgoing tree patterns is equivalent to the summation of the similarities between all the affinal incoming/outgoing tree pattern sets. Then, (14) is rewritten as:

$$k_g^h(G,G') = \sum_{v \in V} \sum_{v' \in V'} k_H(H^h(v), H^h(v')). \tag{20}$$

This kernel is called the tree-pattern graph kernel. Its computational complexity depends on populating $M^-(v,v')$ and $M^+(v,v')$ in (17) and (18). Theoretically, its computational complexity is $O(|V||V'|hB^{2B})$, where B is an upper bound on the incoming degree and the outgoing degree. For applications to action recognition, the graphs are very sparse, resulting in a very small value of B. We design the vertex kernel $k_v()$ and the edge kernel $k_e()$ such that the sizes of the incoming and outgoing neighborhood matching sets are kept small. These designs ensure that the tree-pattern graph kernel is computed efficiently.

The limitation of the tree-pattern graph kernel is that all the similarities between affinal tree-pattern groups are summed up with the same weight. The more discriminative tree patterns are not emphasized.

3.3. Tree-pattern graph matching

To solve the above limitation in the tree pattern graph kernel, we construct a tree-pattern graph matching kernel by finding correctly matched affinal tree-pattern groups using labeled samples.

We assign a weight $\lambda_{v,v'}$ to each pair of affinal tree-pattern groups $H^h(v)$ and $H^h(v')$ from two graphs G and G'. Then, we define the tree-pattern graph matching kernel $k_{mg}^h(G,G')$ for graphs G and G' as follows:

$$k_{mg}(G,G') = \sum_{v \in V} \sum_{v' \in V'} \lambda_{v,v'} k_H(H^h(v), H^h(v')), \tag{21}$$

where $\lambda_{v,v'}$ indicates importance of the matching between $H^h(v)$ and $H^h(v')$.

Besides the measurement $k_H(H^h(v), H^h(v'))$ of the matching quality of a pair of affinal tree-pattern groups $H^h(v)$ and $H^h(v')$, we define a function $c_H(H^h(v), H^h(u), H^h(v'), H^h(u'))$ to measure the pair-wise agreement between pairs $(H^h(v), H^h(v'))$ and $(H^h(u), H^h(u'))$ of affinal tree-pattern groups for vertices v and u in G and vertices v' and u' in G':

$$c_H(H^h(v), H^h(u), H^h(v'), H^h(u')) = k_H(H^h(v), H^h(v')) k_H(H^h(u), H^h(u')) f(v,u,v',u'), \tag{22}$$

where $f(v,u,v',u')$ measures the geometric relations between pairs of affinal tree-pattern groups and is defined according to applications (see Section 4.2). Fig. 5 shows the relations between pairs of affinal tree-pattern groups.

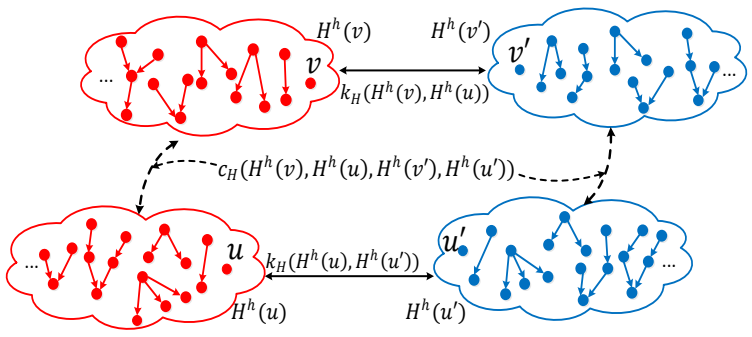


Fig. 5. The relation between two pairs of affinal tree pattern groups: The quality of a matching pair of affinal tree pattern groups $H^h(v)$ and $H^h(v')$ is measured using $k_H(H^h(v), H^h(v'))$, and $c_H(H^h(v), H^h(u), H^h(v'), H(u'))$ measures the pairwise agreement between pairs $H^h(v), H^h(v')$ and $H^h(u), H(u')$.

The correctly matched affinal tree-pattern groups have not only high matching degree, but also high pair-wise agreement. The matched affinal tree-pattern groups should be emphasized and the mismatched affinal tree-pattern groups should be suppressed. Given a pair of samples S and S', graph sets $\{G_l = (V_l, E_l)\}_{l=1}^{L}$, and $\{G'_l = (V'_l, E'_l)\}_{l=1}^{L}$ are extracted respectively. We design a quadratic objective function for S and S' as follows:

$$\max_{\{\lambda\}} \sum_{l} \sum_{v,v'} \lambda_{v,v'} k_H \left(H_l^h(v), H_l^h(v')\right) + \sum_{l} \sum_{v,v'} \sum_{u \neq v, u' \neq v'} \lambda_{v,v'} \lambda_{u,u'} c_H \left(H_l^h(v), H_l^h(u), H_l^h(v'), H_l^h(u')\right) - C \sum_{l} \left(\sum_{v,v'} \lambda_{v,v'}\right)^2 \quad (23)$$
$$s.t. \forall v \in V_l, v' \in V'_l, 0 \leq \lambda_{v,v'} \leq 1,$$

where C is a constant for the $l_{1,2}$-norm sparse constraint for selecting matched affinal tree-pattern groups from the same graphs (the lth graphs) and smoothing the weights of the affinal tree-pattern groups from different graphs (the L graphs for each sample). We determine the weights $\{\lambda_{v,v'}\}$ by solving the quadratic objective function in (23) for two samples efficiently using the trust region reflective algorithm [25]. The obtained weights $\{\lambda\}$ are substituted into (21) to compute the tree-pattern graph matching kernel between samples S and S'. The sparse regularization ensures that only the correctly matched affinal tree-pattern groups with high weights are dominant in the kernel construction. This augments the discriminative ability of the tree-pattern graph matching kernel. The kernel similarities between the training samples are used to train a SVM classifier. Given a test sample, we compute its kernels to the support vectors which are a small subset of the training samples that lie on the maximum margin hyper-planes in the feature space. These kernels are input to the classifier to determine the label of the test sample. The trust region reflective algorithm requires only a few iterations to obtain an effective local solution and a good classification result, even though the algorithm is halted before convergence.

3.4. Relevance to existing kernels

The proposed tree-pattern graph matching kernel has properties from graph kernels and from graph matching kernels. The corresponding analysis is as follows:

The tree-pattern graph matching kernel extends several previous kernels. When all the weights $\{\lambda_{v,v'}\}$ are set to 1, the tree-pattern graph matching kernel reduces to the tree-pattern graph kernel in (20). If we only consider outgoing tree-patterns and set k_v and k_e equal to Dirac kernel functions, then the tree-pattern graph matching kernel reduces to the tree graph kernel in [26]. If parameter γ in (13) is very small, then tree-patterns with a high complexity are penalized, with the result that only tree-patterns consisting of linear chains of vertices have significant weights, i.e., the tree-pattern graph kernel approximates to a traditional random walk graph kernel [20]. If μ in (13) is set to 0, then tree-patterns degenerate into vertices and the tree-pattern graph matching kernel reduces to the summation kernel of vertices [27].

The objective function for determining the weights $\lambda_{v,v'}$ in (23) is related to the graph matching kernel in [28, 29, 30, 31]. In both cases, a unary term and a pair-wise term exploit the local compatibilities and pair-wise geometric relations between substructures of graphs. In contrast with traditional graph matching kernels, our tree-pattern group matching kernel has the following properties:

- The tree-pattern graph matching kernel uses affinal incoming and outgoing tree-pattern groups as substructures to effectively describe the local structure of graphs.

- We add a sparse constraint in (23) to keep well matched affinal tree-pattern sets, and remove noisy ones.

3.5. Comparison with context-dependent random walk graph kernel

We summarize the similarities and differences between the context-dependent random walk graph kernel and the tree-pattern graph matching kernel as follows.

1) Similarities: Both these kernels are constructed by decomposing each graph into sub-graphs and then combining the kernels between the sub-graphs. The sub-graphs into which these two kernels decompose a graph are random walks and tree patterns respectively. These two kernels construct the sub-graph kernels in a similar way: The kernels between sub-graphs are products of the kernels for the vertices and the edges included in the sub-graphs and the graph kernels are sum of the kernels of the sub-graphs into which the graphs are decomposed. They both can be reduced to the traditional random walk graph kernel, as stated in Appendix A and Section 3.4.

2) Differences: Random walks in a graph have simple shapes with chain structures. This limits the ability of random walk kernels to capture sufficient topological information in a graph. Tree patterns have more complex structure and obtain more information about the local topologies of graphs than random walks. A primary walk group is a set of random walks starting at a vertex and ending at another vertex. It describes the local structure as a function of depth, as measured by the edges over which the walk passes. The contexts of a primary walk group supplement the normalized local breadth structure information in the random walk graph kernel. An affinal tree-pattern group is a set of tree-patterns that have the same leaf vertex or the same root vertex. It describes the local structure as a function of both the depth and breadth without normalization. Affinal tree-pattern groups consisting of incoming and outgoing tree-pattern groups describe the local structure both along and against the directions of the edges. Therefore, the tree-pattern graph kernel captures more of the local topological properties of the graphs and more accurately measures similarities of graphs than the context-dependent walk graph kernel. As random walks are more regular, the context-dependent random walk graph kernel can be computed directly and conveniently using the direct product graph. The equation for computing the context-dependent random walk kernel is concise. However, tree-pattern graph matching kernel cannot be computed using the direct product graph. It is recursively computed in a dynamic programming formulation. The recursive solution is elegant. But it requires more runtime than the computation of the context-dependent random walk graph kernel. In addition, the context-dependent random walk graph kernel measures the similarity between two graphs by comparing the pairs of sub-graphs of the graphs, while the tree-pattern graph matching kernel keeps well matched sub-graphs.

4. Action Recognition

We follow the traditional action recognition framework based on points of interest. Given a video containing actions, Dollar's separable linear filters [4] are utilized to detect the spatiotemporal interest points. The 3D SIFT descriptor [17] is used as the local spatiotemporal feature vector to describe each detected interest point. We construct a concurrent graph and a causal graph to model the relations between local feature vectors. Based on the constructed graphs, the similarities between actions are computed for action recognition.

4.1. Graphs for representing actions

The concurrent graph $G_c = (V_c, E_c)$ is constructed to model the spatial relations of interest points in each

frame. Its vertex set V_c consists of the interest points. We employ the ε-graph method to construct the edge set E_c. A sparse affinity matrix $\mathbf{A}_c$ is defined for G_c. Let (x_i, y_i, t_i) be the image and frame coordinates of the *i*th interest point. For two interest points *i* and *j* in the same frame ($t_i = t_j$), if point *j* is up to point *i* in the image and their image distance is close enough, then the element $A_c(i, j)$ in $\mathbf{A}_c$ is 1. In any other case, $A_c(i, j)$ is 0. If $A_c(i, j) = 1$, $(v_i, v_j) \in E_c$. As shown in Fig. 6, G_c is a graph directed from bottom upwards and without loops[1]. The 3D SIFT descriptor of a vertex is used as its attributed feature vector for capturing local appearance information. Attributes are attached to edges according to the relative spatial positions of vertices. The relative spatial position of v_j with regard to v_i can be represented by $\vec{r}(i, j) = (x_j - x_i, y_j - y_i)$. Then, we describe the attribute associated with an edge (v_i, v_j) using $\vec{r}(i, j)$.

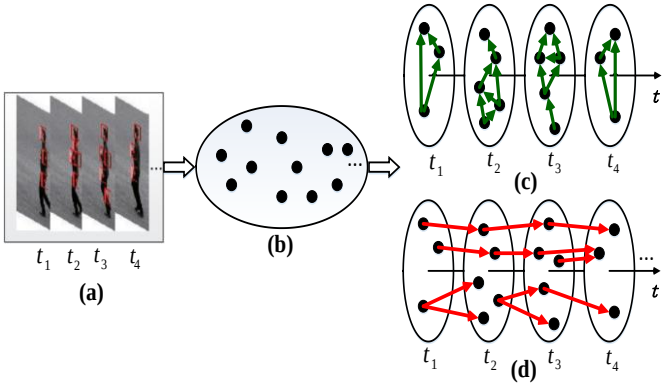


Fig. 6. Graphs for representing actions: (a) A video; (b) Local feature vectors; (c) Concurrent graph; (d) Causal graph.

The causal graph $G_s = (V_s, E_s)$ is constructed to model the spatial relations of local feature vectors between frames. Its vertices V_s as well as the associated attributed vectors are the same as in the concurrent graph. We define an affinity matrix $\mathbf{A}_s$ for G_s. For two interest points *i* and *j* in neighboring frames, if they are close in the image coordinate space, then element $A_s(i, j)$ in $\mathbf{A}_s$ is 1. In any other case, $A_s(i, j)$ is 0. When $A_s(i, j) = 1$, $(v_i, v_j) \in E_s$. As shown in Fig. 6, G_s is a graph directed from left to right. The attribute associated with edge e_{ij} is also described by $\vec{r}(i, j)$. While the directed edge e_{ij} describes a temporal causal relation between v_i and v_j, the assigned edge attribute $\vec{r}(i, j)$ describes their relative spatial position.

The concurrent graph and the causal graph describe different relations between local feature vectors. These two graphs are complementary to preserve the spatiotemporal features of actions.

4.2. Action similarity measurement

We apply the proposed context-dependent random walk graph kernel and tree pattern graph matching kernel to measure the similarity between human actions represented by the concurrent graph and the causal graph. This similarity measurement is based on two basic kernels: the vertex kernel and the edge kernel.

In a concurrent graph, let *d* and d' be the 3D SIFT descriptors for vertices *v* and v', respectively. If the

[1] Any one of the following ways can be used to construct a directed concurrent graph: directed from bottom up, directed from top down, directed from left to right, or directed from right to left. If any two of the four ways are combined, then loops may exist in the graph, and loops may produce traceback in random walks.

distance between d and d' is close in the feature space, then the vertex kernel between them is:

$$k_v(v,v') = \exp\left(-\frac{\|d-d'\|_2^2}{2\sigma^2}\right) \tag{24}$$

where σ is a scale parameter for the Gaussian function. Otherwise, $k_v(v,v')$ is 0. Given the attributes $\vec{r}(i,j)$ of an edge e_{ij} in a concurrent graph, the edge kernel depends on the spatial distance between vertices v_i and v_j and the direction of $\vec{r}(i,j)$. The edge kernel $k_e(e_{ij},e'_{oq})$ between edge e_{ij} in graph G and edge e'_{oq} in graph G' is 1, if vectors $\vec{r}(i,j)$ and $\vec{r}(o,q)$ have similar lengths and directions where the direction similarity is evaluated by the cosine of the angle between $\vec{r}(i,j)$ and $\vec{r}(o,q)$:

$$\frac{\vec{r}(i,j)\cdot\vec{r}(o,q)}{\|\vec{r}(i,j)\|_2\|\vec{r}(o,q)\|_2}. \tag{25}$$

Otherwise, $k_e(e_{ij},e'_{oq})$ is 0.

The vertex kernel for the causal graph is defined in the same way as in the concurrent graph. The edge kernel $k_e(e_{ij},e'_{oq})$ between edge e_{ij} in causal graph G and edge e'_{oq} in causal graph G' is 1, if vectors $\vec{r}(i,j)$ and $\vec{r}(o,q)$ have similar directions, otherwise the edge kernel is 0.

The above definitions of vertex kernels and edge kernels make the two graphs quite sparse. This speeds up the computation process. There are no cycles in either graph. This suppresses the tottering, halting, and backtracking associated with computing the kernels based on random walks or trees.

For the context-dependent random walk graph kernel, based on the defined vertex kernel and edge kernel, the kernels of different orders between the two videos are computed using (7). The $l_{1,2}$-norm regularized generalized multiple kernel learning is used to estimate the weights for the kernels of different orders. Then the similarity between the two videos is computed using (8).

For the tree pattern graph matching kernel, we substitute the defined vertex kernel and edge kernel into (17) and (18) to compute the kernels for the affinal incoming and outgoing tree-pattern groups. Subsequently, we define two variables $d_{ij,oq}$ and $\theta_{ij,oq}$ to describe the relative spatial geometric relations between two vertex pairs $v_i,v_j \in V$ and $v'_o,v'_q \in V'$:

$$\begin{cases} d_{ij,oq} = \|\vec{r}(i,j)-\vec{r}(o,q)\|_2, \\ \theta_{ij,oq} = \arccos\dfrac{\vec{r}(i,j)\cdot\vec{r}(o,q)}{\|\vec{r}(i,j)\|_2\|\vec{r}(o,q)\|_2}. \end{cases} \tag{26}$$

If v_i and v_j are in the same frame in a video and v'_o and v'_q are in the same frame in another video, the geometrical coherence function f in (22) is defined as:

$$f(v_i,v_j,v'_o,v'_q) = \exp\left(-\frac{d_{ij,oq}^2}{2\sigma_d^2}-\frac{\theta_{ij,oq}^2}{2\sigma_\theta^2}\right), \tag{27}$$

where σ_d and σ_θ are scale parameters. Otherwise, f is 0. This definition of f ensures that the affinal tree-pattern group pairs $(H^h(v_i),H^h(v'_o))$ and $(H^h(v_j),H^h(v'_q))$ are used in measuring the pair-wise agreement only when v_i and v_j are in the same frame and v'_o and v'_q are in the same frame. By solving

(23), we obtain the weight for each pair of affinal tree-pattern groups. Then, the tree-pattern graph matching kernel in (21) is computed.

The context-dependent walk graph kernel and the tree-pattern graph matching kernel form a bridge between graphs representing actions and statistic learning methods. Based on the obtained matrix of similarities between videos, the SVM classifier is trained to classify videos containing different actions.

5. Experimental Results

We tested the proposed context-dependent random walk graph kernel-based and tree-pattern graph matching kernel-based action recognition methods on the following benchmark datasets: the Weizman dataset, the KTH dataset [16], the UCF Sports dataset [15], the UCF Films dataset [15], and the Hollywood2 dataset:

- The Weizmann dataset contains 90 videos with 10 actions. Each action was performed by 9 people.
- The KTH dataset has 599 videos, containing six human actions (walking, jogging, running, boxing, hand waving, and hand clapping) which were performed by 25 subjects under four different scenarios.
- The UCF sports dataset consists of 150 broadcast sports videos with ten actions, such as diving, lifting, and running. Videos were captured in realistic scenarios with natural, complex, and cluttered background. The videos exhibit large intra-class variations.
- The UCF films dataset was collected from a range of films genres, such as classic old movies, comedies, and scientific movies. There are 92 samples of kissing and 112 samples of slapping. The actions were captured in a wide range of scenes under different viewpoints.
- The Hollywood2 dataset consists of 12 actions, such as “answer phone” and “drive car”. There are 1707 video samples in total, including 823 samples for training and 884 samples for testing. All the samples were collected from Hollywood movies.

On the Weizman dataset and the KTH dataset, we carried out the leave-one-action-out cross validation to make the performance evaluation, i.e., in each run, the videos of one randomly selected action were used for testing and all the other videos were used for training. On the UCF Sports dataset and the UCF Films dataset, we carried out the leave-one-video-out cross validation. One video was used for testing and the remaining videos were used for training. On the Hollywood2 dataset, the training and test sets were used for training and testing respectively. On the first four datasets, accuracy was used as the evaluation criterion. On the last dataset, the average precision for each action was calculated and the mean average precision on all actions was used as the evaluation criterion.

In the following, we first verify the effectiveness of the graph-based representation for actions. Then, the context-dependent walk graph kernel and the tree-pattern graph matching kernel are compared with their variants to show their effectiveness. Finally, we compare our methods with graph-based methods and state-of-the-art methods for action recognition.

5.1. Effectiveness of graph representation

To illustrate the effectiveness of the concurrent graph and the causal graph for representing actions, we compared our methods with the following variants using the context-dependent random walk graph kernel:

- **The BoW-based method:** A BoW model was used to represent the ensemble of local feature vectors extracted from each video. The *K*-means algorithm was used to cluster the local feature vectors into 1200

visual words. A histogram of words was constructed for each video. A χ^2-kernel was used to measure the similarities between the histograms. An SVM classifier was trained for action classification. The spatiotemporal relations of local features were not involved in this method.

- **The method based on the kernel for attributed point sets:** As stated in Appendix A, the 0th-order context-dependent random walk graph kernel corresponds to the kernel for attributed point sets. This attributed point set kernel was used to measure the similarities between point sets. The obtained kernel matrix was input into an SVM classifier. Although videos were represented by the concurrent graphs and the causal graphs, the edge information in the graphs is not involved in the kernel. Action recognition only depends on the individual discriminative capability of local feature vectors.
- **The concurrent graph-based method:** The ensemble of local feature vectors was modeled by the concurrent graph. The causal graph was omitted. We computed the context dependent graph kernels of different orders on the concurrent graphs and applied the generalized multiple kernel learning to combine these context dependent kernels for action recognition. In this method, the spatiotemporal relations between local feature vectors in the same frames were used.
- **The causal graph-based method:** The ensemble of local feature vectors was modeled by the causal graph. The concurrent graph was omitted. This method uses spatiotemporal relations between local feature vectors in successive frames.

Table 1. The recognition accuracies (%) of our context-dependent random walk graph kernel-based method and the four variants on the Weizmann, KTH, and UCF sports datasets

Methods	Weizamnn	KTH	UCF Sports
BoW-based	92.4	95.0	86.0
Point sets-based	91.4	94.7	85.3
Concurrent graph-based	95.7	95.5	88.7
Causal graph-based	94.6	96.3	89.3
Our context-dependent method	96.9	97.0	90.8

In the experiments, we defined the nearest 5 interest points in the 3D space as the context of a given interest point. We set the maximum order of graph kernels to 5 for both the concurrent graphs and the causal graphs. Table 1 shows the action recognition accuracies of our context-dependent random walk graph kernel method and the above four variants on the Weizmann, KTH, and UCF sport datasets. Due to space limitation, we illustrate the confusion matrices of our context dependent random walk-based method on these three datasets in Fig. A of Appendix E in the supplemental file, which is available online. On analyzing the results, the following points were observed:

- The point sets-based method and the BoW-based method yield the lowest accuracies. This is because these methods only model individual local feature vectors, without considering the spatiotemporal relations between local feature vectors.
- The concurrent graph-based method and the causal graph-based method, which model spatiotemporal relations of local features within a frame and between frames respectively, yield higher accuracies than the point sets-based method and the BoW-based method. This indicates that the spatiotemporal relations preserved in the concurrent graphs and in the causal graphs improve the accuracy of action recognition.
- Our context-dependent walk graph kernel method outperforms both the concurrent graph-based method and the causal graph-based method. This indicates that the concurrent graph and the causal graph, which

capture different spatiotemporal relations, are complementary to each other, and the combination of these two graphs leads a more informative and discriminative representation for human actions.

5.2. Effectiveness of context-dependent graph kernel

To illustrate the effectiveness of the context-dependent walk graph kernel, we compared our method with the variants obtained by replacing the context-dependent walk kernel $k_g^n(G,G')$ in (7) with the traditional random walk graph kernel $k_{tg}^n(G,G')$ shown in Appendix A. In the traditional random walk-based method, the contexts of primary walk groups are not involved in the matching of the primary walk groups, when computing graph kernels. Fig. 7 shows the comparison results on the KTH dataset and the UCF sports dataset when the maximum order was set to 0, 1, 3, 5, and 7, respectively. It is seen that the context-dependent random walk method always yields a higher accuracy than the traditional walk kernel-based method. This is because the proposed context-dependent random walk graph kernel is superior to the traditional random walk kernel in measuring the similarity between graphs. The context information utilized in primary walk group matching improves the performance of action recognition.

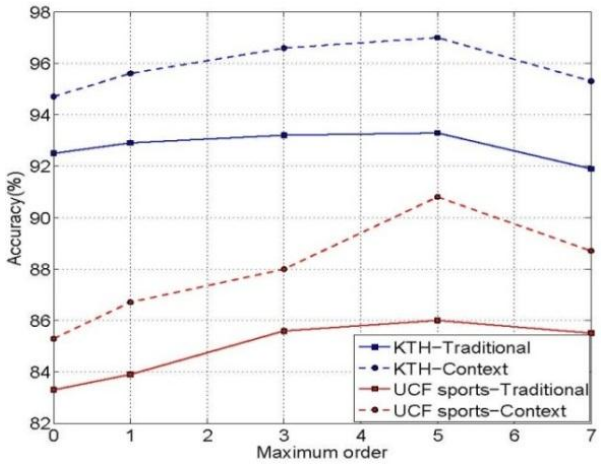


Fig. 7. The results of the traditional random walk kernel and our context-dependent random walk kernel when the maximum order takes values from 0 to 7: the x-axis is the order and the y-axis is the accuracy.

5.3. Effectiveness of tree-pattern graph matching

We evaluated the effectiveness of our tree-pattern graph matching kernel by comparing it with the following variant graph kernels for action recognition:

- **Summation-based graph kernel:** This kernel is defined as the summation [34] of the kernels of all the pairs of vertices from the two graphs. This kernel only considers the individual local features, without taking into account the information of edge attributes and the structures of the graphs.
- **Incoming tree pattern-based graph kernel:** Only the incoming tree pattern-based graph kernel is used to measure the similarity between two graphs. The substructures described by affinal incoming tree-pattern groups were considered for representing the spatiotemporal relations between local feature vectors.
- **Outgoing tree pattern-based graph kernel:** This kernel is computed only based on the affinal outgoing tree-pattern groups. Only outgoing neighborhood information on local feature vectors was considered for measuring their spatiotemporal relations.

- **Tree pattern-based graph kernel:** As shown in (20), this kernel combines the incoming and outgoing tree-pattern graph kernels. It considers both the incoming and outgoing neighborhood information, but all the pairs of tree patterns are assigned the same weight.

Table 2 shows the action recognition accuracies of our tree pattern graph matching-based method and the above four variants on the Weizmann, KTH, and UCF sports datasets. We show the confusion matrices of our tree pattern graph matching-based method on these three datasets in Fig. B of Appendix E in the supplemental file, which is available online. These results illustrate the following interesting points:

- The summation kernel-based method yields the lowest accuracy. This indicates that a simply comparison of the local feature vectors in any two videos is not a sufficient measure of the similarities between videos.
- Both the incoming tree pattern graph kernel and the outgoing tree pattern graph kernel yield more accurate results than the summation graph kernel. This indicates that the local spatiotemporal relations among local feature vectors are of great significance in recognizing actions, and both the incoming and outgoing affinal tree pattern groups improve action recognition.
- The tree pattern-based graph kernel yields higher accuracies than both the incoming tree pattern graph kernel and the outgoing tree pattern graph kernel. This indicates that the combination of incoming and outgoing affinal tree-pattern groups improves the recognition accuracy.
- Our tree pattern graph matching kernel obtains the highest accuracies on these datasets. This indicates that our kernel which incorporates both incoming and outgoing tree-patterns effectively selects the most correctly matched affinal tree-pattern groups and avoids the mismatched ones.
- The tree pattern graph matching kernel is much more accurate than the tree pattern-based graph kernel on the UCF sports dataset in contrast with the Weizmann dataset and the KTH dataset. This is because the UCF sports dataset is much more complex, with occlusions, dynamic backgrounds, and large intra-class variability. The tree pattern graph matching kernel is more suitable and robust for action recognition in complex scenes.

Table 2. The recognition accuracies (%) of our tree pattern graph matching method and the four variant graph kernel-based methods on the Weizmann, KTH, and UCF sport datasets

Methods	Weizmann	KTH	UCF Sports
Summation kernel-based	89.2	92.5	83.3
Incoming tree pattern-based	94.6	95.7	88.6
Outgoing tree pattern-based	92.7	96.1	90.0
Tree pattern-based	95.7	96.3	91.3
Our tree pattern matching-based	97.8	97.2	95.3

We evaluated impact of the height h of the affinal tree-pattern groups on recognition accuracy on the KTH and UCF sports datasets. As shown in Fig. 8, on both the datasets the accuracy increases when h increases from 1 to 4. When h further increases, the accuracy decreases. This is because the affinal tree-pattern groups are appropriate for representing local structures. When the height increases, the leaf vertices of each affinal tree-pattern group spread beyond the local structures. Therefore, the height was set to 4. This is consistent with the viewpoint, in [50], that the spatiotemporal relations among a small number of frames are enough for action recognition. With respect to μ and γ in (13), when their values vary from 1 to 2 the accuracy does not change much. When they are larger or smaller, the accuracy is reduced. Empirically, they were both

set to 1.2.

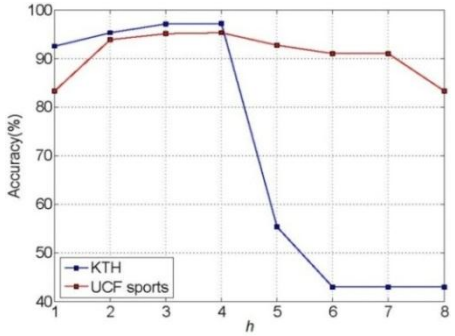


Fig. 8. The accuracies for different values of the height (h) of affinal tree-pattern groups on the KTH and UCF sports datasets.

5.4. Comparison with graph-based methods

We compared our context-dependent random walk graph kernel-based method and our tree pattern graph matching kernel-based method with the previous methods [3, 18, 36, 38, 39, 40] based on graph kernels [36] and graph matching [3, 18]. These algorithms use graphs to represent human actions and apply different graph kernel and graph matching methods for graph comparison. The comparison results on the Weizmann dataset, the KTH dataset, and the UCF sports dataset are shown in Table 3. The following points are noted:

- Overall our methods perform better than the competing graph-based methods [3, 18, 36]. Although our results are slightly lower than the result for the hyper-graph-based method in [18] on the Weizmann dataset, on the KTH dataset our results are much better than those in [18]. This indicates the effectiveness of our graph-based similarity measurements.
- Compared with the random walk graph kernel-based method in [36], our context dependent random walk graph kernel-based method yields a much more accurate result. This partly indicates the effectiveness of our context-dependent random walk graph kernel.
- Our tree pattern graph matching method increases the recognition accuracy on the UCF sports dataset much more than on the Weizmann dataset and the KTH dataset. This indicates the effectiveness of the tree pattern graph matching in adapting to complex scenes.
- Even our tree pattern graph kernel-based method outperforms relatively complex methods such as in [36, 38] on the UCF sports dataset. This demonstrates that the information in the edge attributes and in the tree-structure of graphs is useful for classification.

Table 3. The results (%) of comparison between our methods and previous graph-based methods

Methods	Weizmann	KTH	UCF sports
Ta et al. [18]	100	91.2	-
Celiktutan et al. [3]	-	90.6	-
Wang et al. [36]	-	-	85.2
Jones et al. [38]	-	-	89.1
Guo et al. [39]	-	94.7	-
Ma et al. [40]	-	-	89.4*
Our context-dependent-based	96.9	97.0	90.8
Our tree pattern kernel-based	95.7	96.3	91.3
Our tree pattern matching-based	97.8	97.2	95.3

5.5. Comparison with state-of-the-arts

We compared our context-dependent random walk graph kernel-based method and our tree pattern graph

matching-based method with several state-of-the-art methods on the Weizmann dataset, the KTH dataset, the UCF sports dataset, the UCF films dataset, and the Hollywood2 dataset. The experimental results are shown in Table 4. The following points are observed:

- Our methods overall outperform the listed methods on the KTH dataset, the UCF sports dataset, and the UCF films dataset. In particular, a higher accuracy than the popular deep learning method [45] was obtained. On the UCF sports dataset, the results of our method are, 8.5%, 9.4% and 9.3% higher, respectively, than the latest methods in [41, 42, 44].
- On the Weizmann dataset, out results are less accurate than those obtained by Yeffet et al. [24], but on the KTH dataset, the UCF Sports dataset, and the UCF films dataset, our results are much better than those obtained by Yeffet et al. [24].
- On the Hollywood2 dataset, our methods yield results comparable to the most accurate result in the literature. On this dataset, the state-of-the-art methods are usually based on densely sampled local features, and require a long computational time. Our methods are based on sparse features, but still achieve a good performance. This demonstrates the effectiveness of our graph representation and similarity measurement models.

Table 4. The results (%) of comparison of our methods with the state-of-the-art methods on the five benchmark datasets

Methods, years	Weiz-mann	KTH	UCF Sports	UCF films			Holly-wood2
				Kiss	Slap	Average	
Yeffet et al. [24]	100	90.1	79.2	77.3	84.2	80.7	-
Wang et al. [22]	-	92.1	85.6	-	-	-	47.7
Kovashka et al. [11]	-	94.5	87.3	-	-	-	
Le et al. [12]	-	93.9	86.5	-	-	-	53.3
Junejo et al. [32]	95.3	-	-	-	-	-	-
Wang et al. [21]	-	94.2	88.2	-	-	-	-
Wang et al. [33]	-						58.5
Jiang et al. [10]	-	95.8	88.0	-	-	-	-
Wang et al. [23]	-	93.3	-	-	-	-	-
Celiktutan et al. [3]	-	90.6	-	-	-	-	-
Rodrigues et al. [15]	-	-	-	66.4	67.2	66.8	-
Wang et al. [35]		-	-	86.3	89.6	87.9	-
Zhang et al. [41]	-	94.8	87.5	-	-	-	51.8
Sun et al. [42]	-	93.1	86.6	-	-	-	48.1
Jones et al. [38]	-	-	89.1	-	-	-	59.9
Veeriah et al. [43]	-	-	94.0	-	-	-	-
Wang et al. [44]	-	94.5	86.7	-	-	-	-
Shi et al. [45]	-	95.6	-	-	-	-	-
Gaidon et al. [46]	-	-	-	-	-	-	54.4
Kihl et al. [47]	-	-	-	-	-	-	60.3
Pei et al. [48]	-	-	-	-	-	-	43.9
Gotoh et al. [49]	-	-	-	-	-	-	48.6
Li et al. [60]	-	-	93.4	-	-	-	-
Alfaro et al. [61]	-	97.5	-	-	-	-	-
Our context- dependent-based	96.9	97.0	90.8	97.6	94.4	96.0	58.0
Our tree pattern kernel-based	95.7	96.3	91.3	97.8	94.6	96.2	59.5
Our tree pattern matching-based	97.8	97.2	95.3	97.9	94.7	96.3	60.4

5.6. Comparison between the proposed kernels

The results of comparison between the context-dependent random walk graph kernel and the tree-pattern graph matching kernel on the five benchmark datasets are included in Table 4. The following points are noted:

- On all the five datasets, the tree-pattern graph matching kernel yields more accurate results than the context-dependent random walk graph kernel. This indicates that incorporating both incoming and

outgoing tree-patterns and selecting correctly matched affinal tree-pattern groups is discriminative for action recognition.

- In particular, on the UCF sports dataset, in which the videos show complex scenes, our tree pattern graph matching-based method increases the recognition accuracy by 4.5%, which is a significant improvement over the context-dependent random walk graph kernel. This indicates that the tree pattern graph matching kernel is more suitable for complex scenes.
- On the Weizman dataset, the KTH dataset, the UCF Films dataset, and the Hollywood2 dataset, the improvement of the tree-pattern graph matching kernel over the context-dependent random walk graph kernel is not large. This is because the context-dependent random walk graph kernel already yields state-of-the-art results, which influences the tree-pattern graph matching kernel to yield large improvement over the context-dependent random walk graph kernel. This indicates that the context-dependent random walk kernel and the tree-pattern graph matching kernel are both effective.

6. Conclusion

In this paper, we have proposed a family of context-dependent random walk graph kernels and a family of tree pattern graph matching kernels for the similarity measurement between graphs. In the context-dependent random walk graph kernel, the performance of the primary walk group comparisons is improved by using contexts. The general multiple kernel learning method with the $l_{1,2}$-norm regularization effectively combines context-dependent graph kernels of different orders. In our tree-pattern graph matching kernel, more topological structural information is exploited. We have recursively computed the similarity between affinal tree-pattern groups in a dynamic programming formulation and applied a sparse constraint to match the tree pattern groups. The errors caused by falsely matched affinal tree-pattern groups are suppressed and the discriminative power of the tree pattern graph matching is increased. We have applied the proposed kernels to recognize human actions by constructing the concurrent graph and the causal graph to capture the spatiotemporal relations among local feature vectors. Experimental results on several datasets have demonstrated that the two graphs for representing actions are complementary and the proposed context-dependent random walk graph kernel and tree-pattern graph matching kernel are effective at improving the performance of action recognition. Our tree pattern graph matching kernel yields more accurate results than our context-dependent random walk kernel.

As kernel methods are inefficient, our work is limited to make the experiments on datasets with small numbers of samples. We will investigate to handle this problem in our future work.

References

[1] K.M. Borgwardt, C.S. Ong, S. Schonauer, S. Vishwanathan, A.J. Smola, and H.-P. Kriegel, "Protein function prediction via graph kernels," *Bioinformatics*, vol. 21, pp. 47-56, 2005.

[2] E.Z. Borzeshi, M. Piccardi, and R. Xu, "A discriminative prototype selection approach for graph embedding in human action recognition," *in Proc. of IEEE International Conference on Computer Vision*, pp. 1295-1301, 2011.

[3] O. Celiktutan, C. Wolf, B. Sankur, and E. Lombardi, "Real-time exact graph matching with application in human action recognition," *in Proc. of International Workshop on Human Behavior Understanding*, pp. 17-28, 2012.

[4] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie, "Behavior recognition via sparse spatio-temporal features," *in Proc. of IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pp. 65-72, 2005.

[5] T. Gartner, P. Flach, and S. Wrobel, "On graph kernels: hardness results and efficient alternatives," *Learning Theory and Kernel Machines, vol. 2777 of the series Lecture Notes in Computer Science*, pp. 129-143, 2003.

[6] U. Gaur, Y. Zhu, B. Song, and A. Roy-Chowdhury, "A string of feature graphs model for recognition of complex activities in natural videos," *in Proc. of IEEE International Conference on Computer Vision*, pp. 2595-2602, 2011.

[7] B. Gauzere, L. Brun, D. Villemin, and M. Brun, "Graph kernels based on relevant patterns and cycle information for chemoinformatics," *in Proc. of IEEE International Conference on Pattern Recognition*, pp. 1775-1778, 2012.

[8] Z. Harchaoui and F. Bach, "Image classification with segmentation graph kernels," *in Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1-8, 2007.

[9] W. Imrich and S. Klavzar, "Product graphs: structure and recognition," John Wiley & Sons, New York, 2000.

[10] Z. Jiang, Z. Lin, and L.S. Davis, "Recognizing human actions by learning and matching shape-motion prototype trees," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 34, no. 3, pp. 533-547, 2012.

[11] A. Kovashka and K. Grauman, "Learning a hierarchy of discriminative space-time neighborhood features for human action recognition," *in Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2046-2053, 2010.

[12] Q.V. Le, W.Y. Zou, S.Y. Yeung, and A.Y. Ng, "Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis," *in Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3361-3368, 2011.

[13] M. Parsana, S. Bhattacharya, C. Bhattacharya, and K. Ramakrishnan, "Kernels on attributed pointsets with applications," *in Proc. of Annual Conference on Neural Information Processing Systems*, pp. 1129-1136, 2007.

[14] K. Raja, I. Laptev, P. Perez, and L. Oisel, "Joint pose estimation and action recognition in image graphs," *in Proc. of IEEE International Conference on Image Processing*, pp. 25-28, 2011.

[15] M.D. Rodriguez, J. Ahmed, and M. Shah, "Action MACH: a spatio-temporal maximum average correlation height filter for action recognition," *in Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1-8, 2008.

[16] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing human actions: a local SVM approach," *in Proc. of IEEE International Conference on Pattern Recognition*, pp. 32-36, 2004.

[17] P. Scovanner, S. Ali, and M. Shah, "A 3-dimensional sift descriptor and its application to action recognition," *in Proc. of IEEE International Conference on Microwave Magnetics*, pp. 357-360, 2007.

[18] A.P. Ta, C. Wolf, G. Lavoue, and A. Baskurt, "Recognizing and localizing individual activities through graph matching," *in Proc. of IEEE International Conference on Advanced Video and Signal Based Surveillance*, pp. 196-203, 2010.

[19] M. Varma and B.R. Babu, "More generality in efficient multiple kernel learning," *in Proc. of IEEE International Conference on Machine Learning*, pp. 1065-1072, 2009.

[20] S. Vishwanathan, N.N. Schraudolph, R. Kondor, and K.M. Borgwardt, "Graph kernels," *Journal of Machine Learning Research*, vol. 11, no. 2, pp. 1201-1242, 2010.

[21] H. Wang, A. Klaser, C. Schmid, and C.-L. Liu, "Action recognition by dense trajectories," *in Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3169-3176, 2011.

[22] H. Wang, M. M. Ullah, A. Klaser, I. Laptev, and C. Schmid, "Evaluation of local spatio-temporal features for action recognition," *in Proc. of British Machine Vision Conference*, pp. 124.1-124.11, Sep. 2009.

[23] L. Wang, Y. Qiao, and X. Tang, "Motionlets: Mid-level 3D parts for human motion recognition," *in Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2674-268, 2013.

[24] L. Yeffet and L. Wolf, "Local trinary patterns for human action recognition," *in Proc. of IEEE International Conference on Computer Vision*, pp. 492-497, 2009.

[25] J. Nocdal and S. Wrihgt, "Numerical optimization", In New York; Sringer Verlag, 2nd ed, 2006.

[26] P. Mahe and J.-P. Vert, "Graph kernels based on tree patterns for molecules," *Machine Learning*, vol. 75, no. 1, pp. 3-35, April 2009.

[27] S. Lyu, "Mercer kernels for object recognition with local features," *in Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 223-229, 2005.

[28] M. Cho, J. Sun, O. Duchenne, and J. Ponce. "Finding matches in a haystack: a max-pooling strategy for graph matching in the presence of outliers," *in Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2091-2098, 2014.

[29] T. Cour, P. Srinivasan, and J. Shi, "Balanced graph matching," *in Proc. of Annual Conference on Neural Information Processing Systems*, pp. 313-320, 2006.

[30] A. Egozi, Y. Keller, and G. Hugo, "A probabilistic approach to spectral graph matching," *IEEE Trans. on Patter Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 8-27, 2013.

[31] M. Leordeanu and M. Hebert, "A spectral technique for correspondence problem using pairwise constraints," *in Proc. of IEEE International Conference on Computer Vision*, pp. 1482-1489, 2005.

[32] I.N. Junejo, E. Dexter, I. Laptev, and P. Perez, "View-independent action recognition from temporal self-similarities," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 33, no. 1, pp. 172-185, 2011.

[33] H. Wang and C. Schmid, "Action recognition with improved trajectories," *in Proc. of IEEE International Conference on Computer Vision*, pp. 3551-3558, 2013.

[34] C. Wallraven and B. Caputo, "Recognition with local features: the kernel recipe," *in Proc. of IEEE International Conference on Computer Vision*, pp. 257-264, 2003.

[35] H. Wang, C. Yuan, G. Luo, W. Hu, and C. Sun, "Action recognition using linear dynamic systems," *Pattern Recognition*, vol. 46, no. 6, pp. 710-1718, 2013.

[36] L. Wang and H. Sahbi, "Directed acyclic graph kernels for action recognition," *in Proc. of IEEE International Conference on Computer Vision*, pp. 3168-3175, 2013.

[37] B. Wu, C. Yuan, and W. Hu. "Human action recognition based on context-dependent graph kernels," *in Proc. of IEEE Conference on Computer Vision and Patten Recognition*, pp. 2609-2616, 2014.

[38] S. Jones and L. Shao, "A multigraph representation for improved unsupervised/semi-supervised learning of human actions," *in Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 820-826, 2014.

[39] W. Guo and G. Chen, "Human action recognition via multitask learning base on spatial-temporal feature," *Information Sciences*, vol. 320, no. 3, pp. 418-428, Nov. 2015.

[40] S. Ma, L. Sigal, and S. Sclaroff, "Space-time tree ensemble for action recognition," *in Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5024-5032, 2015.

[41] H. Zhang, W. Zhou, C. Reardon, and L.E. Parker, "Simplex-based 3D spatio-temporal feature description for action recognition," *in Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2067-2074, 2014.

[42] L. Sun, K. Jia, T. Chan, Y. Fang, G. Wang, and S. Yan, "DL-SFA: Deeply-learned slow feature analysis for action recognition," *in Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2625-2632, 2014.

[43] V. Veeriah, N. Zhuang, and G. Qi, "Differential recurrent neural networks for action recognition," *in Proc. of IEEE International Conference on Computer Vision*, pp. 4041-4049, 2015.

[44] D. Wang, Q. Shao, and X. Li, "A new unsupervised model of action recognition," *in Proc. of IEEE International Conference on Image Processing*, pp. 1160-1164, 2015.

[45] Y. Shi, W. Zeng, T. Huang, and Y. Wang, "Learning deep trajectory descriptor for action recognition in videos using deep neural networks," *in Proc. of IEEE International Conference on Multimedia and Expo*, pp. 1-6, 2015.

[46] A. Gaidon, Z. Harchaoui, and C. Schmid, "Activity representation with motion hierarchies," *International Journal of Computer Vision*, vol. 107, no. 3, pp. 219-238, May 2014.

[47] O. Kihl, D. Picard, and P-H. Gosselin, "A unified framework for local visual descriptors evaluation," *Pattern Recognition*, vol. 48, no. 4, pp. 1174-1184, April 2015.

[48] L. Pei, M. Ye, X. Zhao, Y. Dou, and J. Bao, "Action recognition by learning temporal slowness invariant features," *Visual Computer*, vol. 32, no. 11, pp. 1395-1404, Nov. 2016.

[49] N.A. Harbi and Y. Gotoh, "A unified spatio-temporal human body region tracking approach to action recognition," *Neurocomputing*, vol. 161, no. c, pp. 56-64, August 2015.

[50] K. Schindler and L.V. Gool, "Action snippets: how many frames does human action recognition require?" *in Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1-8, 2008.

[51] V. Kantorov and I. Laptev, "Efficient feature extraction, encoding and classification for action recognition," *in Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2593-2600, 2014.

[52] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," *in Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1-8, June 2008.

[53] M. Bregonzio, S. Gong, and T. Xiang, "Recognising action as clouds of space-time interest points," *in Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1948-1955, June 2009.

[54] N.B. Aoun, M. Mejdoub, and C.B. Amar, "Graph-based approach for human action recognition using spatio-temporal features," *Journal of Visual Communication and Image Representation*, vol. 25, no. 2, pp. 329-338, Feb. 2014.

[55] P. Mahe, N. Ueda, T. Akutsu, J.-L. Perret, and J.-P. Vert, "Extensions of marginalized graph kernels," *in Proc. of International Conference on Machine Learning*, pp. 552-559, 2004.

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

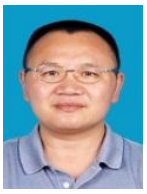
[57] N. Shervashidze, S. Vishwanathan, and T.H. Petri, "Efficient graphlet kernels for large graph comparison," *Journal of Machine Learning Research*, vol. 5, pp. 488-495, 2009.

[58] Y. Kong, Z. Ding, J. Li, and Y. Fu, "Deeply learned view-invariant features for cross-view action recognition," *IEEE Trans. on Image Processing*, vol. 26, no.6, pp. 3028-3037, 2017.

[59] Y. Kong and Y. Fu, "Max-margin heterogeneous information machine for RGB-D action recognition," *International Journal of Computer Vision*, vol. 123, no.3, 350-371, 2017.

[60] Q. Li, H. Cheng, Y. Zhou, and G. Huo, "Human action recognition using improved salient dense trajectories," *Computational Intelligence and Neuroscience*, vol. 2016, Article ID: 6750459.

[61] A. Alfaro, D. Mery, and A. Soto, "Action recognition in video using sparse coding and relative features," *in Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2688-2697, 2016.

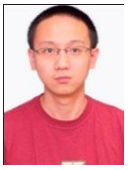


Weiming Hu received the Ph.D. degree from the department of computer science and engineering, Zhejiang University in 1998. From April 1998 to March 2000, he was a postdoctoral research fellow with the Institute of Computer Science and Technology, Peking University. Now he is a professor in the Institute of Automation, Chinese Academy of Sciences. His research interests are in visual motion analysis, recognition of web objectionable information, and network intrusion detection.

Baoxin Wu received the B.S. degree in automation from the Ocean University of China, QingDao, China in 2010, and the Ph.D. degree in Pattern Recognition and Intelligent System in Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2015. He is currently a research assistant at Sogou, Beijing, China. His current research interests mainly focus on computer vision.

Pei Wang received the B.E. degree in measurement control technology and instrument from the University of Electronic Science and Technology of China, Chengdu, China, in 2014. He is currently a graduate student pursuing the Master's degree with National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China. His current research interests include computer vision and machine learning.

Chunfeng Yuan received the doctoral degree from the Institute of Automation, Chinese Academy of Sciences (CASIA), Beijing, China, in 2010. She is currently an associate professor at the CASIA.. Her research interests and publications range from statistics to computer vision, including sparse representation, motion analysis, action recognition, and event detection.



Yangxi Li is a senior engineer of National Computer network Emergency Response technical Team/Coordination Center of China (CNCERT/CC). He received the Ph.D. degree from Peking University. His research interests lie primarily in multimedia search, information retrieval and computer vision.

Stephen Maybank received a BA in Mathematics from King's college Cambridge in 1976 and a PhD in computer science from Birkbeck college, University of London in 1988. Now he is a professor in the School of Computer Science and Information Systems, Birkbeck College. His research interests include the geometry of multiple images, camera calibration, visual surveillance etc.