Temporal Activity Graph Kernel based Human Activity Classification

ABSTRACT

Human activities are predominantly spatio-temporal, involving spatial changes over time. Qualitative spatial relations between interacting entities are often used to describe spatial change. An *extended object* based abstraction of spatial entities has been shown to be more effective in [11]. The temporal aspect of the activity is characterized through changing spatial relations between components of interacting *extended* objects over time. In this paper, we propose the use of Temporal Activity Graphs (TAG) for keeping track of the sequences of relations between components of the extended objects. We present a *kernel* designed for classification of spatio-temporal interactions represented as a TAG. This kernel is used within a Support Vector Machine classifier. Experiments are performed on the Mind's Eye, the UT Interaction, and the SBU Kinect Interaction datasets. The TAG kernel based classification of activities is found comparable to *state-of-the-art* approaches.

KEYWORDS

kernel, temporal activity graph, human activity classification, SVM, qualitative relations, Extended CORE9

ACM Reference Format:

. 1997. Temporal Activity Graph Kernel based Human Activity Classification. In *Proceedings of ICVGIP (ICVGIP'18).* ACM, New York, NY, USA, 8 pages. https://doi.org/10.475/123_4

1 INTRODUCTION

Recognition of activities within a video is a subject of great attention because of its applications in various interesting problems such as automated patient monitoring, ambient assisted living, automated surveillance, and content-based retrieval. Human activity recognition (HAR) focuses on activities of human which are more complex and varied. HAR involves automated learning of interaction models, i.e. general description of the interactions that can be used for recognizing similar activities [7]. A sequence of frames for an instance of the *handshake* activity from the UT Interaction dataset [17] is shown in Fig. 1. Learning an interaction model for *handshake* would allow a HAR system to recognize the handshake activity in any video thereafter.

A proper abstraction of human bodies affects efficiency and effectiveness in a HAR system. It has been shown that an extended object representation is appropriate for human activities [11]. An activity is characterized by the temporal evolution of spatial relations between components of entities. A graph-based representation is capable of adequately encoding temporal information. In this work, we represent an activity as a *Temporal Activity Graph* (TAG). Nodes in the graph correspond to components of interacting objects; labels on edges describe the *qualitative spatial relations* between the components at a specific time point. The set of *sequences* of edge labels over different time points express the relational changes between the objects during the activity. When human bodies are abstracted as *extended objects* [10], the graph structure allows a neat way of keeping track of how the spatial relations between the body parts evolve. Additionally the graph structure can be conveniently used in several learning techniques such as, support vector machines, graph based relational learning, graph grammar induction among others. We propose a *kernel function* for activities represented as TAGs, so that a Support Vector Machine (SVM) can be used for classification of the activities. The proposed TAG *kernel* computes similarity of two activities based on concepts of *interestingness* and *similarity* of label sequences.

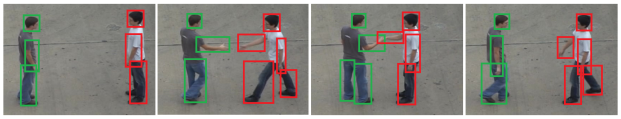

Figure 1: *Handshake* from the UT-Interaction Dataset

2 LITERATURE REVIEW

Representation of human activities using graphs have been reported in literature by various researchers [1, 9, 20, 24]. The temporal nature of activities has often been modeled using probabilistic graphical models such as HMMs [1, 9]. However, such models do not explicitly record or exploit the spatial relations or features of objects that change over time during an activity.

Researchers have also used graphical models that encode spatial and temporal features of an activity simultaneously [20, 24]. To address the problems of using only spatial or only temporal features, researchers have represented activities as temporal sequences of *structured feature graphs* [24]. The correlation between spatial features in a single frame of the video have been modeled using Conditional Random Fields. Such structured feature graphs are computed for every frame of the video and temporally sequenced to encode the spatio-temporal nature of an activity. However, temporal sequencing of disjoint graphs do not allow keeping track of how individual spatial features evolve over time. Activities represented as hierarchical qualitative spatio-temporal graphs have also been reported in literature [20]. The researchers have encoded qualitative spatial relations between objects as vertices at one level and qualitative temporal relations as vertices at a higher level. However, in their work the researchers abstract individual interacting entities using a single bounding box. Using such a representation for an extended object based abstraction will lead to an explosion

ICVGIP'18, December 2018, Hyderabad, India

ACM ISBN 123-4567-24-567/08/06…$15.00

https://doi.org/10.475/123_4

in the number of vertices in the graph. Temporal Activity Graph representation discussed herein is designed for extended objects.

More recent research in HAR has largely focused on deep learning based approaches [14, 21, 28]. 3D Convolutional Neural Networks [21]; Recurrent Neural Network [14]; Recurrent Neural Networks with Long Short-Term Memory [28] have been quite successful. Such techniques fail to take into account high-level information [27]. High-level relational information as encoded in TAGs could give interesting insights for learning interaction models from relatively small data-sets.

2.1 Preliminaries of Qualitative Reasoning Formalisms

Qualitative Spatio-Temporal Reasoning (QSTR) provides formalisms for capturing common-sense knowledge pertaining to space and time [4]. QSTR formalisms have often been used by researchers for a more intuitive description of video activities [5, 7, 20].

Two common aspects of space that are often discussed within QSTR are topology and direction. Topology is concerned with relations that are not affected by any change in size and/or shape of the objects. Region Connection Calculus (RCC) is one of the most well-known frameworks for expressing qualitative topological relations between a pair of regions, that have co-dimension zero [16]. Based on a topological primitive of *connection*, eight base relations referred to as RCC8 relations are defined. The RCC8 relations are Disconnected (DC), Externally Connected (EC), Partially Overlapping (PO), Equal (EQ), Tangential Proper Part (TPP) and its inverse (TPPI), and Non-Tangential Proper Part(NTPP) and its inverse (NTTPI). The DC and EC relations of RCC8 can be abstracted by a single Disjoint (DR) relation; TPP and NTPP relations are abstracted as Proper Part (PP); and TPPI and NTPPI relations are abstracted as Proper Part Inverse (PPI) leading to RCC5.

Qualitative size relations have been defined using a order-of-magnitude reasoning that deals with primitive relations *same-size-as* and *roughly-the-same-size-as* [2]. The qualitative distance relations are defined within a mereological framework of part-whole relations together with qualitative size relations and a *connection* primitive, *sphere* primitive. The *sphere* primitive is used to define the region within which a particular distance relation holds. This framework defines qualitative distance relations, such as Close (Cl), Strictly Close (SCl), Near (N), Strictly Near (SN), Away (A), Far Away (FA), Moderately Away (MA), for a pair of disconnected sphere regions. An approximation of qualitative distance relation for a pair of extended objects, (approx-QD) has been discussed within the Extended CORE9 framework [11]. Directional relations of a pair of regions are expressed as one of the eight cardinal directions: North (N), NorthEast (NE), East (E), SouthEast (SE), South (S), SouthWest (SW), West (W), NorthWest (NW) - or some combination of these [18].

2.1.1 *Conceptual Neighbourhood Graph*.

A *conceptual neighbourhood graph* (CNG) can be defined for the set of base relations. The nodes of the CNG correspond to a single qualitative relation. An edge between two nodes (say between nodes R_1 and R_2) indicate that a direct *transition* from R_1 to R_2 is possible [4]. In this context, a direct *transition* from R_1 to R_2 indicates that if the relation R_1 holds between two entities at a given time point, then R_2 may hold between the time entities in the immediate next time point, by continuous transformation of the entities [8]. CNGs are important for reasoning in a qualitative framework. We use CNG to compute the similarity of a sequence of relations between a pair of interacting entities (See Sec. 4.2).

2.2 Extended CORE9

CORE9 is an integrated representation that allows encoding of several interesting spatial information between a pair of rectangles [6]. CORE9 captures topological, directional, size, distance, and motion related information in a compact form. Extended CORE9 is an extension of CORE9 that was proposed to compute spatial information between a pair of *extended objects* [11]. Therein, extended objects have been defined as *a set of components, where each component is approximated by an axis-aligned minimum bounding rectangle*. It has been argued that extended objects provide a better abstraction of human bodies for the purpose of HAR. Binary spatial relation corresponding to a pair of extended objects, is expressed in the form of *component* relations and *whole* relations. Component relations are the spatial relations between components of one extended object with components of the other. Whole relations capture the general spatial relations between the extended objects. The component and whole relations are computed opportunistically using a general recursive algorithm. Within the Extended CORE9 framework, computation of topological, directional and distance relations for a pair of extended objects has been discussed. We use component relations thus computed to represent an activity.

3 REPRESENTATION OF VIDEOS AS TAGS

We represent an activity as a TAG that captures the sequence of relations between the interacting objects. In a video depicting an activity, we abstract each object as an *extended object* [11]. Each frame corresponds to a specific time point during the activity.

Definition 3.1. An **activity** is defined as a set of sequences of component relations and whole relations between a pair of extended objects over a set of time points.

This is based on the intuition that change of relations between extended objects over the sequence of time points is fairly distinctive for every activity. We represent such an activity using a *temporal graph* as shown in Fig. 2.

Definition 3.2. A **temporal activity graph** G is formally denoted by a 5-tuple $(X, \mathcal{T}, V, E_s, E_t)$, where,

- $X = A \cup B \cup \ldots$, where $A, B, \ldots$ are any number of **extended objects involved in the activity** such that $A = \{a_i | i = 1 \ldots n$ and n is the number of components in $A\}$, $B = \{b_j | j = 1 \ldots m$ and m is the number of components in $B\}$ and so on.
- $\mathcal{T} : X \times \mathbb{N} \rightarrow \{0, 1\}$ is the **time function**. Here, $\mathcal{T}(a_i, t) = 1$ iff component a_i appears in the video activity at time point t. Here, a_i is a component of the extended object A.
- $V = \{a_i^t | a_i \in X$ and $\mathcal{T}(a_i, t) = 1\}$ is the **set of vertices**.
- $E_s = \{(a_i, b_j, t) \mid a_i, b_j \in X$ and $\mathcal{T}(a_i, t) = \mathcal{T}(b_j, t) = 1\}$ is the **set of directed spatial edges**. Further, an edge label is associated with each spatial edge that is denoted by $\varepsilon(a_i, b_j, t)$.

- $E_t = \{(a_i, t, t+k) \mid a_i \in X,\ \mathcal{T}(a_i,t) = \mathcal{T}(a_i,t+k) = 1,\ \mathcal{T}(a_i,t+1) = \ldots = \mathcal{T}(a_i,t+k-1) = 0\}$ is the **set of directed temporal edges**

Semantically, $\mathcal{T}(a_i,t) = 1$ iff A appears in the video at frame number t.

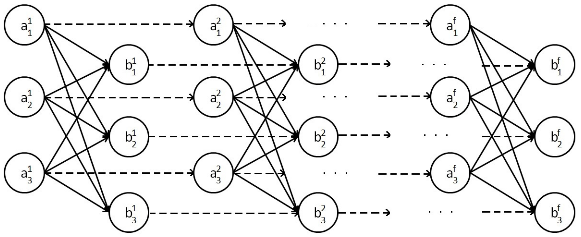

Figure 2: A temporal graph representing an video activity consisting of f frames

In Fig. 2 an activity involving two extended objects, A and B, is depicted as a TAG; here, A and B have components a_1, a_2, a_3 and b_1, b_2, b_3 respectively. The solid edges correspond to *spatial edges* and are labeled with the spatial relations between the respective components at the specific time point.

Definition 3.3. The **edge label** between a_i^t and b_j^t, denoted by $\varepsilon(a_i, b_j, t)$, is a three tuple $\langle top - dir - dis \rangle$ where *top* is the topological relation, *dir* is the directional relation and *dis* is the distance relation between a_i^t and b_j^t.

The dashed links correspond to temporal changes and appear between a_i^t and a_i^u, such that component a_i appears in the video at time point t and reappears at some subsequent time point u $(t < u)$. If t and u are not consecutive then a_i does not appear in the video at any time point between t and u. Thus, by traversing along the temporal edges, it is possible to give a description of how the spatial relation (obtained from the solid edge labels) between components a_i and b_j changes over the entire duration of the activity as recorded in the video. Such a description can be given as a sequence of edge labels and we term it a *label sequence*.

Definition 3.4. In a TAG, G, a **label sequence** $ls_{i,j}$ between components a_i and b_j is the sequence of edge labels $\langle \varepsilon(a_i, b_j, 1), \varepsilon(a_i, b_j, 2), \ldots \varepsilon(a_i, b_j, t) \rangle$.

The label sequence between a pair of components describe how the spatial relations between them change over time. If at any time point t, either of the components a_i and b_j do not appear in the video, then the corresponding edge label, $\varepsilon(a_i, b_j, t)$ is replaced by a NULL value in the label sequence $ls_{i,j}$. Every activity is characterized by the set of label sequences of the corresponding TAG. Similarity between two activities can be established by comparing the respective sets of label sequences (see Sec. 4).

In the following section, we illustrate the concepts of TAG, label-sequences and edge-labels using the sample activity sequence in Fig. 1.

3.1 Illustrative Example

Let us consider the sequence of three video frames obtained from a sample *Handshake* activity in the UT Interaction dataset [17], as shown in Fig. 1. The activity sample involves two human bodies A and B; each human body is an extended object of at most five components. The TAG for the activity is shown in Fig. 3. In the figure, components of A are labeled a_x^t, where x refers to the component identifier and t refers to the corresponding time point; similarly components of B are labeled b_x^t. Here, $x \in \{h, rh, lh, rl, ll\}$; h refers to *head*, rh refers to *right hand*, lh refers to *left hand*, rl refers to *right leg* and ll refers to *left leg*.

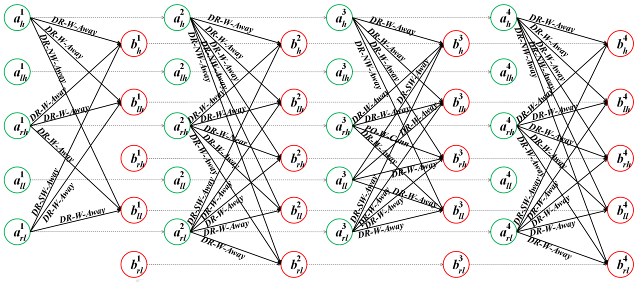

Figure 3: TAG for the sequence of frames in Fig. 1 that corresponds to *Handshake* activity from the UT-Interaction Dataset

In the TAG shown in Fig. 3, the spatial edge labels are 3-tuples of topological, directional and distance relations computed using Extended CORE9 [11]. In the frames where a particular component does not appear due to occlusion, the edges corresponding to that component are not drawn. For every pair of component of A and B, we obtain the label sequence, i.e. the sequence of edge labels for $t = 1, 2, 3, 4$. In this example, the label sequence for the components a_{rh} and b_{rh} is: $\langle DR - W - Away \rangle, \langle DR - W - Near \rangle, \langle PO - W - Conn \rangle, \langle DR - W - Away \rangle$. The edge labels between a_h and b_h does not change over time; the label-sequence is $\langle DR - W - Away \rangle$. The component a_{lh} does not appear in any of the time points, therefore all label sequences between components a_{lh} and b_x is *NULL*.

4 A KERNEL FOR TEMPORAL ACTIVITY GRAPHS

Kernel methods implicitly define possible patterns by introducing a notion of similarity between data. For linearly separable data, the similarity between data is computed as the inner product of the feature vectors. In some cases, the data may not be linearly separable; the data is then transformed into a non-linear feature space using some function ϕ, wherein the data is linearly separable and similarity can be computed using the inner product of the transformed feature vectors. However, instead of mapping the data to an implicit feature space and then computing the similarity using inner product, it is enough to define a single *kernel* function to compute the similarity of the data in the implicit feature space. For a function to qualify as a *kernel* function that can be used in a SVM, it has to exhibit properties similar to inner product in some implicit feature space. In this section, we introduce a *temporal activity graph kernel* and show that this kernel is *symmetric* and *positive semi definite* in Theorem 4.4.

We define a TAG kernel that computes a real number signifying the similarity between a pair of TAGs. In order to compute the similarity of two temporal activity graphs G and G', we consider the set of label sequences for each graph. The similarity of the two TAGs is computed as the similarity of the sets of label sequences that characterize the TAGs. In other words, the set of label sequences of G is compared to the set of label sequences of G'.However, if every label sequence of G is compared to every label sequence of G', then the number of label sequence comparison will be $O(n^2)$ where n is the number of label sequences in G or G'. Such an exhaustive comparison is not only computationally expensive but also ineffective; it disregards the fact that a label sequence is between a pair of distinct components and usually is not the same as a label sequence between a different pair of components. Instead, a one-to-one comparison between the sets of label sequences requires fewer computations and is more effective.

In order to perform a one-to-one comparison of the sets of label sequences, it is necessary to identify which label sequence of one set will be compared which label sequence of the other set. Therefore, we assign an intrinsic order to the components of an entity in the TAG. In Eqn. 1, to perform one-to-one comparison, we compute the similarity of label sequences ls_{ij} and ls'_{pq} only if the intrinsic order of a_i in G is the same as intrinsic order of a'_p in G' and the order of b_j in G is the same as the order of b'_q in G'.

We have experimented with two different intrinsic order of the components.

- *Based on Skeletal Information*: Certain part-based tracking systems allow identification of the skeletal structures of human bodies being tracked [13, 15]. In such cases the components can have a fixed order based on which part of the skeletal structure they correspond to.
- *Based on Interestingness*: Components of an entity can be ordered based on an *interestingness factor* or *i-factor*. The *i-factor* captures how involved a component is in the activity. For example, a component corresponding to the *hand* will be more involved in a *handshake* activity but will be less involved in a *kick* activity. The computation of the *i-factor* is discussed in Sec. 5.2.

Definition 4.1. The **temporal activity graph kernel** is defined as $\kappa : \mathcal{G} \times \mathcal{G} \rightarrow \mathbb{R}$, where $\mathcal{G}$ is the set of temporal activity graphs and $\mathbb{R}$ is the set of real numbers, such that,

$$\kappa(G, G') = \frac{c}{1 + \sum\limits_{i,j,p,q} \kappa_{ls}(ls_{ij}, ls'_{pq})} \tag{1}$$

Here, c is a constant computed from the maximum number of components for an extended object in the system, the function κ_{ls} computes similarity of a pair of label sequences.

4.1 Label Sequence Similarity

The similarity of a pair of label sequences is defined as a modified *edit distance*. The Wagner-Fischer algorithm to compute edit distance finds the minimum number of *editing* operations (viz. *insert*, *delete*, or *substitute*) required to transform one string to another [22]. Given two strings a and b, an alignment of the strings involves finding a way of lining up the characters of a and b including mismatches and gaps. If there is mismatch then the edit operation required to match the strings is *substitution*; if there is a gap in a then the edit operation required is *deletion*; if there is a gap in b then the edit operation required is insertion. Usually, insertion and deletion operations have cost 1 and substitution operation has cost 2. The strings are aligned such that the total cost of the edit operations is the smallest.

To compute similarity of a pair of label sequences, we use the Wagner-Fischer algorithms using modified costs. The cost of edit operations depends on the similarity of edge labels, as described in Eqn. 4. The algorithm uses dynamic programming to compute $d_{u,t}$ for a pair of label sequences $a = \langle e_a^1, e_a^2, \ldots e_a^u \rangle$ and $b = \langle e_b^1, e_b^2, \ldots e_b^t \rangle$. Here, u and t are the lengths of the label sequences a and b, that may or may not be equal. We use the following recurrence with modified costs.

$$d_{i,0} = i \tag{2}$$

$$d_{0,j} = j \tag{3}$$

$$d_{i,j} = \begin{cases} d_{i-1,j-1} & \text{if } e_a^i = e_b^j \\ min \begin{Bmatrix} d_{i-1,j} + 1 - \kappa_{edge}(e_a^{i-1}, e_b^j) \\ d_{i,j-1} + 1 - \kappa_{edge}(e_a^i, e_b^{j-1}) \\ d_{i-1,j-1} + 1 - \kappa_{edge}(e_a^i, e_b^j) \end{Bmatrix} & \text{otherwise} \end{cases} \tag{4}$$

The similarity of the label sequences is then defined as:

$$\kappa_{ls}(a, b) = d_{u,t} \tag{5}$$

Here, the function $\kappa_{edge}(ex, ey)$ computes the similarity of the edge labels ex and ey. The similarity function κ_{edge} is discussed in the following section.

4.2 Edge Label Similarity

We have defined an edge label to be a three-tuple of topological, direction and distance relation (Def. 3.3).Topological relations are expressed as RCC5 relations [16], directional relations are expressed as Cardinal Direction Calculus relations [18], and distance relations are an approximation of Qualitative Distance relations [2]. These relations are computed using Extended CORE9 [10]. To compute similarity, we define *neighbourhood distance* for a pair of qualitative relations within *conceptual neighbourhood graph*(CNG) (see Sec. 2.1.1).

An edge in the CNG between a pair of relations indicate that a direct transition from one relation to the other is possible. The similarity between a pair of relations R_1 and R_2 can be expressed as a function of the number of direct transitions required to go from R_1 to R_2.

Definition 4.2. **Neighbourhood based similarity** ($\mathcal{N}_s^{C^Q}$) between a pair of relations R_1 and R_2 using the conceptual neighbourhood graph C^Q, for some qualitative relational calculus Q, is defined as,

$$\mathcal{N}_s^{C^Q}(R_1, R_2) = 1 - \frac{p^{C^Q}(R_1, R_2)}{p_{max}^{C^Q}} \tag{6}$$

Here, $p^{C^Q}(R_1, R_2)$ is the length of the shortest path between R_1 and R_2 in C^Q and $p_{max}^{C^Q}$ is the maximum length of a shortest path between any pair of relations in C^Q.

Definition 4.3. The **similarity of a pair of edge labels** $\varepsilon_1 = \langle top_1, dir_1, dis_1\rangle$ and $\varepsilon_2 = \langle top_2, dir_2, dis_2\rangle$ is defined as the weighted average of the *neighbourhood distances* of the topological, directional and distance relations.

$$\begin{aligned}\kappa_{edge}(\varepsilon_1, \varepsilon_2) =& w_1 * \mathcal{N}_s^{C^{RCC5}}(top_1, top_2)\\ &+ w_2 * \mathcal{N}_s^{C^{CDC}}(dir_1, dir_2)\\ &+ w_3 * \mathcal{N}_s^{C^{QD}}(dis_1, dis_2) \qquad (7)\end{aligned}$$

where $w_1 + w_2 + w_3 = 1$ and C^{RCC5}, C^{CDC}, and C^{CDC} are the CNG for Region Connection Calculus, Cardinal Direction Calculus, and Qualitative Distance Relations respectively.

The values of the neighbourhood based similarity index lie in the range $[0, 1)$. As a result, the values of the edge label similarity function lies between 0 and 1.

Lemma 4.4. *The values computed by edge label similarity function, κ_{edge}, lies between 0 and 1, where 1 denotes exactly similar edge labels.*

Proof. In Eqn. 6, $p^{C^Q}_{max} \geq p^{C^Q}(R_1, R_2)$. Therefore, $\mathcal{N}_s^{C^Q}(R_1, R_2)$ is a value that lies between 0 and 1. The value of κ_{edge} as computed by Eqn. 7, is the weighted average of $\mathcal{N}_s^{C^{RCC5}}$, $\mathcal{N}_s^{C^{CDC}}$ and $\mathcal{N}_s^{C^{QD}}$ such that the sum of the weights is 1. Hence, values computed by edge label similarity function, κ_{edge}, lies between 0 and 1. □

Theorem 4.5. *The* temporal activity graph kernel*, κ, (in Eqn. 1) is symmetric and positive semi-definite.*

Proof. *Temporal activity graph kernel* κ (Eqn. 1) depends on the label sequence similarity function, κ_{ls} (Eqn. 5) and the edge label similarity function κ_{edge} (Eqn. 7).

By definition the *neighbourhood based similarity* (Def. 4.2) of spatial relations is a symmetric function. This ensures κ_{edge}, the weighted average of three neighbourhood based similarity values, is symmetric. The value computed by the label sequence similarity function κ_{ls}, is a modified *edit distance* of the label sequences. Traditionally, edit distance is a symmetric measure because the cost of complementary *insert* and *delete* operations are the same. In our case, the modified edit distance computed is symmetric because the cost of the complementary *insert* and *delete* operations is symmetric and the minimum value amongst the three edit costs does not change. The label sequence similarity is symmetric therefore the sum of label sequence similarity values is symmetric. The *temporal activity graph kernel* is symmetric.

From Lemma 4.4, κ_{edge} for exactly similar pair of edge labels is 1; for all other possible pairs the value is in the range $[0, 1)$. This is reversed in Eqn 5 - for exactly similar label sequences the modified edit distance is 0; for all other possible pairs, the value is greater than 0. Using this Eqn. 5 in Eqn 1 ensures that for a pair of activities that are exactly similar the *kernel* value computed is the largest. If K is the kernel matrix such that $K_{ij} = \kappa(G_i, G_j)$ then $K_{ii} > K_{ij} \forall i \neq j$. Thus, K is a diagonally dominant matrix. It is a property of diagonally dominant matrices that they are positive definite i.e. the *temporal activity graph kernel* is positive semi definite. □

5 CLASSIFICATION OF HUMAN ACTIVITIES

To perform classification of human activities, we first obtain part-based tracking data of the activities in the video. The entities involved in the activity are abstracted as *extended objects*; for every frame of the video, qualitative spatial relations between extended objects are computed using Extended CORE9 [10]. The spatio-temporal knowledge thus obtained about the activity are represented within a *temporal activity graph*.The spatial relations computed using the Extended CORE9 framework are used as *edge labels* for the *spatial edges* in the graph. The activities represented as temporal activity graphs are then classified using a Support Vector Machine (SVM) based on the kernel function defined in Eqn 1.

In Eqn 1, the kernel value for two activities represented as graphs G and G' is computed as the sum of similarity of *label sequence* computed using Eqn 5. The similarity of the label sequences is computed on a one-to-one basis, i.e. every label sequence of G is matched with exactly one label sequence of G' and vice versa. In order to determine which pair of label sequences from G and G' are compared, the components are assigned an intrinsic order based on: 1. *skeletal structure* and 2. *interestingness*.

5.1 Using Skeletal Information

It is possible to track the pose of the human body in video [13, 15]. In such part-based tracking, the various human body parts are tracked individually to give a more accurate estimation of the human pose at any given point of time. Tracking a single human body gives a sequence of locations of each individual body-part. In other words, such tracking systems allow the human body to be viewed as an *extended object*. Further, it is possible to label each component based on which body it corresponds to.

In our first approach, we order the components based on the labels of such a part based tracking system. For example, say the tracking system tracks the body parts: *head*(h), *right hand* (rh), *left hand* (lh), *right leg* (rl), and *left leg* (ll). The components for a tracked human body can be ordered as $\langle h, rh, lh, rl, ll\rangle$.With respect to Eqn 1, for two activities represented as G and G', such an order defines that the similarity of label sequence between heads of the interacting human of G ($ls_{h,h}$) and the label sequence between the heads of the interacting humans in G' ($ls'_{h,h}$) is computed. Similarly $ls_{h,ll}$ is compared with $ls'_{h,ll}$, $ls_{rh,rh}$ is compared with $ls'_{rh,rh}$ and so on. The sum of the all the similarity values computed for such one-to-one pairs of label sequences (as described in Sec. 4.1) is the kernel value of the corresponding *temporal activity graphs*.

5.2 Using Interestingness

In our second approach, we consider the case when explicit labels for the components are not available. In this case, the components of the extended objects are ordered based on an *interestingness factor* or *i-factor*. The *i-factor* is computed based on a component's involvement within an activity. To determine how involved a component is in a particular activity we use the intuitive notion that - *a component of an entity that is more involved in the activity will have more number of spatial relational changes with components of the other entity.* The *i-factor of a component* a_i (written as $\mathcal{I}_c(a_i)$) is defined as the sum of the *i-factors* of the label sequences a_i is associated with (written as $\mathcal{I}_l(ls_{i,p})$) where $ls_{i,p}$ is a label sequence between components a_i and b_p). Here a_i and b_p are components of entities A and B respectively, involved in an activity. Eqns 8 and 9 are used to compute *i-factor* of a component and *i-factor* of a label sequence respectively.

$$\mathcal{I}_c(a_i) = \sum_{p=1}^{n} \mathcal{I}_l(ls_{i,p}) \tag{8}$$

$$\mathcal{I}_l(ls_{i,p}) = \sum_{u=1}^{v-1} \kappa_{edge}(\varepsilon(a_i, b_p, t_u), \varepsilon(a_i, b_p, t_{u+1})) \tag{9}$$

In Eqn 9, we assume that label sequence $ls_{i,p}$ is $\langle \varepsilon(a_i, b_p, 1), \varepsilon(a_i, b_p, 2), \ldots\ \varepsilon(a_i, b_p, v)\rangle$ for the activity graph G. Since we are working on human interactions, in Eqn 8, we assume involvement of two entities (A and B with m and n components respectively).

Using Eqn 8, we compute the *i-factor* for all components of all entities. For each entity the components are ordered in decreasing order of their *i-factor*. We apply this order to compute the kernel value of two activity graphs, G and G', using Eqn 1. Thus the label sequence between components with highest *i-factor* of both entities in G is matched with label sequence between components with highest *i-factor* of both entities in G' and so on.

5.3 Experimental Evaluation

Experiments were performed using 110 videos from the Mind's Eye [1], 50 videos from the UT-Interaction [17] dataset and 282 videos from the SBU Kinect Interaction dataset [25]. For the Mind's Eye dataset the following 11 activities are considered- *approach, carry, catch, collide, drop, follow, hold, kick, pickup, push* and *throw* - and 10 videos for each activity. For the UT Interaction dataset we consider five activities that involve at least two humans- *handshaking, hugging, kicking, punching* and *pushing*. Since tracks for the UT-Interaction and Mind's Eye dataset, are not available, humans or objects involved in the activities are manually labeled for each keyframes in the video [2]. The SBU Kinect Interaction dataset consists of eight activities over 282 videos and skeleton tracks for all the videos are available. We use all activities for this dataset and the available tracks are used to obtain the extended object representation.

For every video activity, we obtain the TAG as described in the previous sections. The edge labels are obtained using the Extended CORE9 framework [11]. The TAG-*kernel* based SVM is used for classification. Results for both *skeletal information* based TAG kernel and *interestingness* based TAG kernel are reported. The *precision*, *recall* and *f1-score* are computed for each dataset; the results for the Minds's Eye dataset are given in Table. 1, results for the UT Interaction dataset are given in Table. 2 and results for the SBU Kinect Interaction dataset in Table. 3.

Activity	Skeletal Information			I-factor		
	P	R	F1	P	R	F1
Approach (10)	0.50	0.60	0.55	0.57	0.40	0.47
Carry (10)	1.00	0.70	0.82	1.00	0.60	0.75
Catch (10)	0.91	1.00	0.95	0.89	0.80	0.84
Collide (10)	0.47	0.80	0.59	0.67	0.20	0.31
Drop (10)	0.64	0.70	0.67	0.63	0.50	0.56
Follow (10)	0.88	0.70	0.78	0.78	0.70	0.74
Hold (10)	0.73	0.80	0.76	0.48	1.00	0.65
Kick (10)	0.86	0.60	0.71	0.57	0.80	0.67
Pickup (10)	1.00	1.00	1.00	0.91	1.00	0.95
Push (10)	0.86	0.60	0.71	0.46	0.60	0.52
Throw (10)	0.67	0.60	0.63	0.67	0.60	0.63

Table 1: Results for TAG Kernel based SVM Classification on Mind's Eye dataset

Activity	Skeletal Information			Interestingness		
	P	R	F1	P	R	F1
Handshaking(10)	1	1	1	0.82	0.9	0.86
Hugging (10)	0.91	1	0.95	0.77	1	0.87
Kicking (10)	1	0.8	0.89	1	0.9	0.95
Punching (10)	0.75	0.9	0.82	0.60	0.6	0.60
Pushing (10)	0.89	0.8	0.84	0.86	0.6	0.71

Table 2: Results for TAG Kernel based SVM Classification on UT Interaction dataset

Activity	Skeletal Information			Interestingness		
	P	R	F1	P	R	F1
Approaching (42)	0.97	0.83	0.90	0.86	0.74	0.79
Departing (43)	0.76	0.86	0.80	0.73	0.81	0.77
Pushing (40)	0.86	0.78	0.82	0.86	0.75	0.80
Kicking (41)	0.74	0.78	0.76	0.76	0.76	0.76
Punching (18)	0.88	0.83	0.86	0.76	0.72	0.74
Exchanging(21)	0.95	0.90	0.93	0.86	0.86	0.86
Hugging(39)	0.67	0.85	0.75	0.67	0.82	0.74
Handshaking(38)	0.94	0.78	0.85	0.83	0.79	0.81

Table 3: Results for TAG Kernel based SVM Classification on SBU Kinect Interaction dataset

Table 4 shows that good classification accuracy is obtained for all the considered datasets, when the skeletal structure is used for comparing the label sequence sets. We have also experimented by considering only topological relation, only directional relation, only distance relation and a combination of the three as the edge labels on the TAG. It has been noted that if only the topological relation is used, then the classification accuracy is higher than any other combination for all the datasets. Further, the lowest classification accuracy is obtained when only the directional relation is used. This is because, for most activities, the cardinal direction between the interacting entities may change based on the angle of viewing. The classification accuracies obtained when using a combination of the three relations is also brought down because of this reason.

In the absence of the skeletal structure information, using the proposed *interestingness* factor for matching pair of label sequences (i.e. the interestingness based TAG kernel) also works reasonably well. Table. 4 shows that the classification accuracies in this case are lower than the skeletal information based TAG kernel. This is expected because in *skeletal information based kernel*, additional information is available. Whereas in case of *interestingness based kernel* the absence of skeletal information is dealt with by using a heuristic to match the label-sequences i.e. *i-factor*. Other than

[1] www.visint.org

[2] We use I-frames obtained using the tool *ffmpeg* as keyframes, www.ffmpeg.org

Method	UTI	ME	SBUKI
ExtCORE9 BoW + KNN [11]	62%	58.18%	40.78%
ExtCORE9 BoW + SVM [11]	74%	55.45%	47.16%
ExtCORE9 BoW + Naive Bayes [11]	74%	55.45%	49.64%
ExtCORE9 BoW + Deep Learning [11]	64%	57.27%	46.45%
TAG (only Topological) + Skeletal Information	90%	73.63%	81.91%
TAG (only Directional) + Skeletal Information	72%	50.00%	59.92%
TAG (only Distance) + Skeletal Information	82%	63.63%	67.73%
TAG (Topological + Directional + Distance) + Skeletal Information	78%	64.54%	72.69%
TAG (only Topological) + Interestingness	80%	65.45%	78.01%
TAG (only Directional) + Interestingness	62%	44.54%	51.41%
TAG (only Distance) + Interestingness	72%	61.81%	65.24%
TAG (Topological + Directional + Distance) + Interestingness	68%	60.91%	71.27%

Table 4: Comparison of classification accuracies on the UT Interaction (UTI) dataset, Mind's Eye (ME) dataset and SBU Kinect Interaction (SBUKI) dataset

Method	UTI	ME	SBUKI
TAG + Skeletal Information based Kernel	90%	73.63%	81.91%
TAG + Interestingness based Kernel	80%	65.45%	78.01%
ExtCORE9 BoW + Naive Bayes [11]	74%	55.45%	49.64%
ExtCORE9 BoW + Deep Learning [11]	64%	57.27%	46.45%
Angled CORE9 + LDA [19]**	-	64.4%	-
BoW + SVM [26]	77%	-	-
BoP + SVM [26]	95%	-	-
Skeleton + Deep LSTM [28]	-	-	86.03%

Table 5: Comparison of classification accuracies with other approaches in literature. (In [19] only 5 activities of the Mind's Eye Dataset are considered.)**

that, the results are similar. In the *interestingess* based TAG kernel too, the highest accuracy is obtained when using only topological relations and the lowest accuracy is obtained when using only directional relations. Table 5 shows a comparison of classification accuracies obtained using our work and existing literature. In case of Mind's Eye dataset, our work outperforms Angled CORE9 with a bag-of-words representation [19], despite the fact that we use 11 activity classes whereas [19] experiments on only five activities. We have computed the average MCC for the *skeletal information based kernel* in Mind's Eye dataset is 0.7 which is again higher than what is reported for [3] (0.37). However, we use only 11 activity classes, whereas [3] uses all activity classes in the Mind's Eye dataset. We have also achieved reasonably good accuracy for the SBU Kinect Interaction dataset and UT Interaction dataset even though it does not surpass the state-of-the-art results. It is to be noted that most state-of-the-art work in the field of human activity recognition has been achieved using data driven algorithms and deep learning. Even though such approaches often give high classification accuracy, they often require large training sets and high computation power devices. In our work we focus on the high level knowledge obtained from the tracking data, that allows one to learn richer models from smaller datasets and do not require high computation on power devices.

6 CONCLUSION

In this paper we present *temporal activity graph* (TAG) - an extension of temporal graph - to represent activities recorded in video. Such a representation is capable of keeping track of the changing spatial relations between objects and their components over the duration of the activity. In order to enable classification of activities using such a graph structure, a *temporal activity graph kernel* is defined. The kernel value for a pair of activities represented as TAGs is the sum of similarity of *label sequences*. The similarity of the label sequences is computed on a one-to-one basis; where components are assigned an intrinsic order based either *skeletal structure* or *interestingness*.

In literature, there exists graph based representations that model only the temporal nature of activities [1, 9] or only the spatial features of the video activities [23]. In contrast, the TAG representation models spatial and temporal features of activities within the same graph structure. Other graph representations that model spatial and temporal features simultaneously have been reported in literature [20, 24]. Unlike TAG, such representations either do track the temporal evolution of spatial features [24], or do not scale well with extended object based abstraction [20].

For learning from graph representations of human activities, graph based relational learning [20] and graph kernels [24] have been reported in literature among others. However, graph kernels are a more generic solution that can be used with any kernel based method [12]. There exist graph based approaches that encode either only spatial features or only temporal structure. The TAG kernel defined herein directly takes advantage of the temporal structure of activities in computing the similarity of graphs.

Evaluation is done over videos from Mind's Eye dataset, UT Interaction dataset, and SBU Kinect Interaction dataset. The results obtained using the kernel within an SVM for classification of activities in the datasets show that the kernel gives good precision and recall values for a majority of the activity classes. However, for certain activities with similar relational changes between components, the kernel is not able to provide a well-defined boundary. Investigations towards a more logic-based learning mechanism to eliminate such cases is part of ongoing research. Other possible research directions include identification of a minimum cardinality label sequence set for better interpretation of activities.

REFERENCES

[1] M. Ahmad and Seong-Whan Lee. 2006. HMM-based Human Action Recognition Using Multiview Image Sequences. In *18th International Conference on Pattern Recognition (ICPR)*. IEEE, 263–266. https://doi.org/10.1109/ICPR.2006.630

[2] Thomas Bittner and Maureen Donnelly. 2007. A formal theory of qualitative size and distance relations between regions. In *Proc. of the 21st Annual Workshop on Qualitative Reasoning (QR 2007)*.

[3] Henri Bouma, Gertjan Burghouts, Leo de Penning, Patrick Hanckmann, Johan-Martijn ten Hove, Sanne Korzec, Maarten Kruithof, Sander Landsmeer, Coen van Leeuwen, Sebastiaan van den Broek, Arvid Halma, Richard den Hollander, and Klamer Schutte. 2013. Recognition and localization of relevant human behavior in videos. In *Sensors, and Command, Control, Communications, and Intelligence (C3I) Technologies for Homeland Security and Homeland Defense XII*. SPIE, 87110B–10. https://doi.org/10.1117/12.2015877

[4] A. G. Cohn and S. M. Hazarika. 2001. Qualitative Spatial Representation and Reasoning: An Overview. *Fundamenta Informaticae* 46, 1-2 (2001), 1–29.

[5] Anthony G. Cohn, Derek R. Magee, Aphrodite Galata, David Hogg, and Shyamanta M. Hazarika. 2003. Towards an Architecture for Cognitive Vision Using Qualitative Spatio-temporal Representations and Abduction. In *Spatial Cognition*. Springer Berlin Heidelberg, 232–248. https://doi.org/10.1007/3-540-45004-1_14

[6] Anthony G. Cohn, Jochen Renz, and Muralikrishna Sridhar. 2012. Thinking Inside the Box: A Comprehensive Spatial Representation for Video Analysis. In *Proc. 13th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR2012)*. AAAI Press, 588 – 592.

[7] Krishna S. R. Dubba, Mehul Bhatt, Frank Dylla, David C. Hogg, and Anthony G. Cohn. 2012. Interleaved Inductive-Abductive Reasoning for Learning Complex Event Models.. In *ILP (Lecture Notes in Computer Science)*, Vol. 7207. Springer, 113–129.

[8] Christian Freksa. 1992. Temporal Reasoning Based on Semi-intervals. *Artificial Intelligence* 54, 1-2 (1992), 199–227. https://doi.org/10.1016/0004-3702(92)90090-K

[9] M. Humayun Kabir, M. Robiul Hoque, Keshav Thapa, and Sung-Hyun Yang. 2016. Two-Layer Hidden Markov Model for Human Activity Recognition in Home Environments. *International Journal of Distributed Sensor Networks* 12, 1 (2016), 12. https://doi.org/10.1155/2016/4560365

[10] Shobhanjana Kalita, Arindam Karmakar, and Shyamanta M. Hazarika. 2017. Comprehensive Representation and Efficient Extraction of Spatial Information for Human Activity Recognition from Video Data. In *Proceedings of International Conference on Computer Vision and Image Processing: CVIP 2016, Volume 2*. Springer Singapore, 81–92. https://doi.org/10.1007/978-981-10-2107-7_8

[11] Shobhanjana Kalita, Arindam Karmakar, and Shyamanta M. Hazarika. 2018. Efficient extraction of spatial relations for extended objects vis-à-vis human activity recognition in video. *Applied Intelligence* 48, 1 (2018), 204–219. https://doi.org/10.1007/s10489-017-0970-8

[12] Pierre Latouche and Fabrice Rossi. 2015. Graphs in machine learning: An introduction. In *23rd European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*. 207–218.

[13] Vlad I. Morariu, David Harwood, and Larry S. Davis. 2013. Tracking People's Hands and Feet Using Mixed Network AND/OR Search. *IEEE Transactions on Pattern Recognition and Machine Intelligence (PAMI)* 35, 5 (2013), 1248–1262.

[14] S.U. Park, J.H. Park, M.A. Al-masni, M.A. Al-antari, Md.Z. Uddin, and T.-S. Kim. 2016. A Depth Camera-based Human Activity Recognition via Deep Learning Recurrent Neural Network for Health and Social Care Services. *Procedia Computer Science* 100 (2016), 78 – 84.

[15] V. Ramakrishna, T. Kanade, and Y. Sheikh. 2013. Tracking Human Pose by Tracking Symmetric Parts. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. 3728–3735. https://doi.org/10.1109/CVPR.2013.478

[16] David A. Randell, Zhan Cui, and Anthony Cohn. 1992. A Spatial Logic Based on Regions and Connection. In *KR'92. Principles of Knowledge Representation and Reasoning: Proc. of the 3rd Int. Conf.*, Bernhard Nebel, Charles Rich, and William Swartout (Eds.). Morgan Kaufmann, 165–176.

[17] M. S. Ryoo and J. K. Aggarwal. 2010. UT-Interaction Dataset, ICPR contest on Semantic Description of Human Activities (SDHA). http://cvrc.ece.utexas.edu/SDHA2010/Human_Interaction.html. (2010).

[18] Spiros Skiadopoulos and Manolis Koubarakis. 2005. On the consistency of cardinal directions constraints. *Artificial Intelligence* 163 (2005), 91 – 135.

[19] Hajar Sadeghi Sokeh, Stephen Gould, and Jochen J. 2013. Efficient Extraction and Representation of Spatial Information from Video Data.. In *Proc. of the 23rd Int. Joint Conf. on Artificial Intelligence (IJCAI'13)*. AAAI Press/IJCAI, 1076–1082.

[20] Muralikrishna Sridhar, Anthony G. Cohn, and David C. Hogg. 2010. Relational Graph Mining for Learning Events from Video. In *5th Starting AI Researchers Symposium (STAIRS)*. 315–327. https://doi.org/10.3233/978-1-60750-676-8-315

[21] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. 2015. Learning Spatiotemporal Features with 3D Convolutional Networks. In *2015 IEEE International Conference on Computer Vision (ICCV)*. 4489–4497.

[22] R. Wagner and M. Fischer. 1974. The String-to-String Correction Problem. *J. ACM* 21, 1 (1974), 168 – 173.

[23] Y. Wang and G. Mori. 2011. Hidden Part Models for Human Action Recognition: Probabilistic versus Max Margin. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)* 33, 7 (2011), 1310–1323. https://doi.org/10.1109/TPAMI.2010.214

[24] W. Xu, Z. Miao, and X. P. Zhang. 2015. Structured feature-graph model for human activity recognition. In *IEEE International Conference on Image Processing (ICIP)*. IEEE, 1245–1249. https://doi.org/10.1109/ICIP.2015.7350999

[25] Kiwon Yun, Jean Honorio, Debaleena Chattopadhyay, Tamara L. Berg, and Dimitris Samaras. 2012. Two-person Interaction Detection Using Body-Pose Features and Multiple Instance Learning. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*. IEEE.

[26] Yimeng Zhang, Xiaoming Liu, Ming-Ching Chang, Weina Ge, and Tsuhan Chen. 2012. Spatio-Temporal Phrases for Activity Recognition. In *Computer Vision – ECCV 2012: 12th European Conference on Computer Vision, Proceedings, Part III*. Springer Berlin Heidelberg, 707–721. https://doi.org/10.1007/978-3-642-33712-3_51

[27] Yibiao Zhao, Steven Holtzen, Tao Gao, and Song-Chun Zhu. 2015. Represent and Infer Human Theory of Mind for Human-Robot Interaction. In *2015 AAAI Fall Symposium Series*.

[28] Wentao Zhu, Cuiling Lan, Junliang Xing, Wenjun Zeng, Yanghao Li, Li Shen, and Xiaohui Xie. 2016. Co-Occurrence Feature Learning for Skeleton Based Action Recognition Using Regularized Deep LSTM Networks. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, 2016*. 3697–3704.