Chapter 4

Graph-based Representation and Classification of Activities

4.1 Introduction

In Chapter 3, we have presented Extended CORE9, a framework for efficiently computing topological, direction and distance relationships between a pair of *extended objects*. Extended CORE9 enables one to arrive at the qualitative spatial relations between the entities involved in an activity, abstracted as extended objects. These qualitative spatial relations describe the spatial configuration of the entities at a specific instant of time during the activity. However, activities are spatio-temporal in nature. An activity is characterized by the evolution of spatial relations between entities over time. In this chapter, we present *Temporal Activity Graphs* that can keep track of how relations between entities, abstracted as extended objects, change over time.

In literature, *temporal graphs* have been defined as tools that are used to describe events over periods of time [85]. Further, temporal graphs have the analytical advantage of being static graphs while retaining all temporal information that may be available to us. In this chapter, a *Temporal Activity Graph* representation is presented that allows keeping track of how spatial relations between body parts evolve during a human activity. We propose a *kernel function* for activities represented as Temporal Activity Graphs, so that a support vector machine (SVM) can be used for classification of the activities. The proposed *Temporal Activity Graph Kernel* computes similarity of two activities based on *label sequence similarity*, *edge label similarity*, *neighbourhood-based similarity* and *interestingness factor* introduced herein.

4.1.1 Desiderata for Temporal Activity Graphs

Representation of human activities using graphs have been reported in literature by various researchers [45, 46, 49, 86, 87]. The temporal nature of activities has often been modeled using probabilistic graphical models such as HMMs [49, 86]. However, such models do not explicitly record or exploit the spatial relations or features of objects that change over time during an activity. On the other hand, graphical models like Hidden Conditional Random Fields have also been used to correlate spatial features of the video activities [51]. These models do not encode the temporal structure of activities.

Researchers have also used graphical models that encode spatial and temporal features of an activity simultaneously [46, 87]. To address the problems of using only spatial or only temporal features, researchers have represented activities as temporal sequences of *structured feature graphs* [46]. 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 [87]. 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 in the number of vertices in the graph. Spatio-temporal graph representations have also been designed based on a volumetric video representation wherein vertices are spatio-temporal segments of the video [45]. Such approaches rely on spatial as well as temporal segmentation of the video.

Temporal activity graphs presented in the following section was developed with the aim of addressing the limitations of existing graph representations for human activities.

4.2 Temporal Activity Graphs

An activity represented as a *Temporal Activity Graph* captures the sequence of relations between the interacting entities. In a video depicting an activity, we abstract each object as an *extended object*. Each video frame corresponds to a specific time point during the activity. Thus, an activity can be seen as the evolution of spatial relations between the pair of extended objects over a set of time points. Based on the intuition that change of relations between extended objects over the sequence of time points is fairly distinctive for every activity, we define an *activity* as follows.

Definition 4.1. *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.*

In literature, temporal graphs have been used to describe events that take place over time [85]. Vertices correspond to objects at a specific instant of time. Edges can be of two types: (a) Edges corresponding to relations between the entities at the same instant of time (b) Edges corresponding to temporal evolution of an entity over time. In our work, we abstract entities (humans or objects involved in the activity) as extended objects. Therefore, we define *Temporal Activity Graphs* wherein *vertices* correspond to components of the extended objects at discrete time points over the duration of the activity. *Edges* in a Temporal Activity Graph are of two types:

(a) Edges between vertices at the same time point are used to describe spatial relation of the activity at that instant of time; such edges are termed *spatial edges*. Spatial edges appear only between components of different extended objects.

(b) Edges between vertices at different time points are used to describe temporal evolution of a component; such edges are termed *temporal edges*. A sequence of frames, where each *frame* is a snapshot taken at a specific *time point*. In this thesis, the terms *frames* and *time points* are used interchangeably.

In Figure 4-1, an activity involving two extended objects, A and B, is depicted as a *Temporal Activity Graph*; here, A and B have components a_1, a_2, a_3 and b_1, b_2, b_3 respectively. We use the notation a_i^t to denote a vertex in the Temporal Activity Graph, that correspond to component a_i at a given time point / frame, t. Similarly, b_j^t refers to the vertex corresponding to component b_j at time t. The solid edges correspond to *spatial edges* which are labeled with the qualitative spatial relations between the respective components at the specific time point. The dashed links correspond to *temporal edges* 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

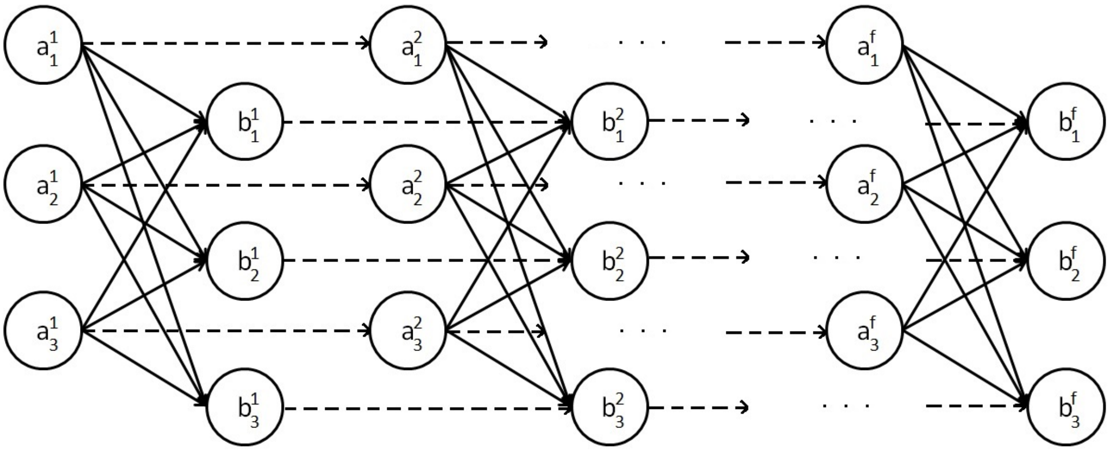

Figure 4-1: A Temporal Activity Graph representing an video activity consisting of f frames

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 spatial edge labels) between components a_i and b_j changes over the duration of the activity. Such a description can be given as a sequence of edge labels and we term it a *label sequence*.

Given Definition 4.1, a temporal graph representation of activities termed *Temporal Activity Graph* (TAG), is defined in Definition 4.2. The terms *edge labels* and *label sequences* with respect to TAGs are also defined below.

Definition 4.2. *A* ***Temporal Activity Graph*** *G is formally denoted by a 5-tuple $(X, \mathcal{T}, V, E_s, E_t)$, where,*

- *$X = A \cup B \cup ...$, where $A, B, ...$ are any number of* ***extended objects involved in the activity*** *such that*
  $A = \{a_i | i = 1...n$ and n is the number of components in $A\}$,
  $B = \{b_j | j = 1...m$ 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 $\varepsilon(a_i, b_j, t)$.*

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

Definition 4.3. *The* ***edge label*** *between a_i and b_j at time point 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 and b_j at time t.*

Definition 4.4. *In a TAG, G, a* ***label sequence*** *$ls^{u:t}_{i,j}$ $(u < t)$ between components a_i and b_j is the sequence of edge labels $\langle \varepsilon(a_i, b_j, u), \varepsilon(a_i, b_j, u+1), ...\varepsilon(a_i, b_j, t) \rangle$.*

The label sequence between a pair of components, from two different extended objects, describe how the spatial relations between them change over time. If at any time point v $(u \leq v \leq t)$, either of the components a_i and b_j do not appear in the video, then the corresponding edge label, $\varepsilon(a_i, b_j, v)$ is replaced by a NULL value in the label sequence $ls^{u:t}_{i,j}$.

Every activity can be described using a *Temporal Activity Graph* (TAG). It follows from Definition 4.1 that, a TAG is characterized by a set of vertices corresponding to components of extended objects at discrete time points, and by the set of spatial and temporal edges between the vertices. The spatial edges between a pair of components are labeled and the sequence of edge labels between a pair of components over time is a label sequence. Thus, every activity is characterized by the set of label sequences of the corresponding TAG. similarity of two activities can be established by comparing the respective sets of label sequences.

4.3 Temporal Activity Graph Kernel

In this section a TAG kernel is defined, that computes a real number signifying the similarity of a pair of TAGs. To compute similarity of two Temporal Activity Graphs X and Y, 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. The set of label sequences of X is compared to the set of label sequences of Y. However, if every label sequence of X is compared to every label sequence of Y, then the number of label sequence comparison will be $O(n^2)$ where n is the number of label sequences in X or Y. Such an exhaustive comparison is computationally expensive. Instead, if a label sequence of a TAG is compared to exactly one label sequence of the second TAG (a one-to-one comparison), then a fewer computations will be required. Furthermore, an exhaustive comparison is ineffective because it disregards the fact that a label sequence between a pair of distinct components is not likely to be the same as a label sequence between a different pair of components.

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 to which label sequence of the other set. Therefore, we assign an *intrinsic order* to the components of an entity in the TAG; a more detailed discussion of intrinsic order is given in Section 4.3.3. Let us consider two TAGs X and Y, corresponding to extended objects A, B and C, D respectively. X is defined over u time points and Y is defined over t time points. The label sequence between a pair of components a_i and b_j in X is written as $x_{ij}^{1:u}$; similarly, $y_{pq}^{1:t}$ is the label sequence between components c_p and d_q in Y. In Equation 4.1, to perform one-to-one comparison, we compute the similarity of label sequences $x_{ij}^{1:u}$ and $y_{pq}^{1:t}$ only if the intrinsic order of a_i in X is the same as intrinsic order of c_p in Y and the intrinsic order of b_j in X is the same as the intrinsic order of d_q in Y.

We have experimented with two different intrinsic order of the components (see Section 4.3.3):

- *Based on Skeletal Information*: Certain part-based tracking systems allow identification of the skeletal structures of human bodies being tracked [119, 120]. 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.

Definition 4.5. *The* ***Temporal Activity Graph Kernel*** *is defined as* $\kappa : \mathcal{G} \times \mathcal{G} \to \mathbb{R}$*, where* $\mathcal{G}$ *is the set of Temporal Activity Graphs and* $\mathbb{R}$ *is the set of real numbers, such that,*

$$\kappa(X, Y) = \frac{c}{1 + \sum\limits_{i,j,p,q} \kappa_{ls}(x_{i,j}^{1:u}, y_{p,q}^{1:t})} \tag{4.1}$$

Here, c *is a constant computed from the maximum number of components for an extended object in the system,* u *is the number of time points in* X*,* t *is the number of time points in* Y*, and the function* κ_{ls} *computes similarity of a pair of label sequences.*

4.3.1 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 [121]. 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 Equation 4.4. The algorithm uses dynamic programming to compute $d_{u,t}$ for a pair of label sequences $a^{1:u} = \langle e_a^1, e_a^2, ...e_a^u\rangle$ and $b^{1:t} = \langle e_b^1, e_b^2, ...e_b^t\rangle$. 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.

$$d_{i,0} = i \tag{4.2}$$

$$d_{0,j} = j \tag{4.3}$$

$$d_{i,j} = \begin{cases} d_{i-1,j-1} & \text{if } e_a^i = e_b^j \\ min \left\{ \begin{array}{l} d_{i-1,j} + 1 - \kappa_{edge}(e_a^{i-1}, e_b^j) \\ d_{i,j-1} + 1 - \kappa_{edge}(e_a^i, e_b^{j-1}) \\ d_{i-1,j-1} + 1 - \kappa_{edge}(e_a^i, e_b^j) \end{array} \right\} & \text{otherwise} \end{cases} \tag{4.4}$$

The similarity of the label sequences is then defined as:

$$\kappa_{ls}(a^{1:u}, b^{1:t}) = d_{u,t} \tag{4.5}$$

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.

4.3.2 Edge Label Similarity

The edge labels in a TAG specify the spatial relation between a pair of components. From Definition 4.3, an edge label is a three-tuple of topological, direction and distance relation and are computed using Extended CORE9. Topological relations are expressed as RCC5 relations [70], directional relations are expressed as Cardinal Direction Calculus relations [75], and distance relations are an approximation of Qualitative Distance relations [72].

The similarity of a pair of edge labels can be computed as the weighted average of the similarities of topological, direction and distance relations. To compute similarity of a pair of qualitative relations, we define a *neighbourhood-based similarity* based on the respective *conceptual neighbourhood graphs* (CNGs). A conceptual neighbourhood graph (CNG) for a qualitative calculus is a directed graph with vertices corresponding to the base relations. An edge between two nodes (say between vertices R1 and R2) indicate that a direct transition from R1 to R2 is possible. An edge in the CNG between a pair of relations indicate that a direct transition from one relation to the other is possible (see Section 2.4.1 in Chapter 2). We express the similarity of a pair of relations R_1 and R_2 as a function of the number of direct transitions required to go from R_1 to R_2. The *neighbourhood-based similarity* and similarity of edge labels are defined as follows.

Definition 4.6. ***Neighbourhood based similarity*** *($\mathcal{N}_s^{\mathcal{C}^Q}$) between a pair of relations R_1 and R_2 using the conceptual neighbourhood graph $\mathcal{C}^Q$, for some qualitative relational calculus Q, is defined as,*

$$\mathcal{N}_s^{\mathcal{C}^Q}(R_1, R_2) = 1 - \frac{p^{\mathcal{C}^Q}(R_1, R_2)}{p_{max}^{\mathcal{C}^Q}} \tag{4.6}$$

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

Definition 4.7. *The* ***similarity of a pair of edge labels*** *$e_1 = \langle top_1, dir_1, dis_1\rangle$ and $e_2 = \langle top_2, dir_2, dis_2\rangle$ is defined as the weighted average of the neighbourhood based similarities of the topological, directional and distance relations.*

$$\begin{aligned}\kappa_{edge}(e_1, e_2) =& w_1 * \mathcal{N}_s^{\mathcal{C}^{RCC5}}(top_1, top_2)\\ &+ w_2 * \mathcal{N}_s^{\mathcal{C}^{CDC}}(dir_1, dir_2)\\ &+ w_3 * \mathcal{N}_s^{\mathcal{C}^{QD}}(dis_1, dis_2)\end{aligned} \tag{4.7}$$

where $w_1+w_2+w_3 = 1$ and $\mathcal{C}^{RCC5}$, $\mathcal{C}^{CDC}$, and $\mathcal{C}^{CDC}$ are the CNG for Region Con-

nection Calculus, Cardinal Direction Calculus, and Qualitative Distance Relations respectively (see Figure2-7).

Lemma 4.1. *The values computed by the neighbourhood based similarity function lie in the range* $[0, 1]$*.*

Proof. In Equation 4.6, $p^{\mathcal{C}}_{max} \geq p^{\mathcal{C}}(R_1, R_2)$. Therefore, $\mathcal{N}^{\mathcal{C}}_s(R_1, R_2)$ is a value that lies between 0 and 1. □

Lemma 4.2. *The values computed by edge label similarity function,* κ_{edge}*, lie in the range* $[0, 1]$*, where 1 denotes exactly similar edge labels.*

Proof. The value of κ_{edge} as computed by Equation 4.7, is the weighted average of $\mathcal{N}^{\mathcal{C}^{RCC5}}_s$, $\mathcal{N}^{\mathcal{C}^{CDC}}_s$ and $\mathcal{N}^{\mathcal{C}^{QD}}_s$ such that the sum of the weights is 1. Further, from Lemma 4.1, neighbourhood based similarity values lie in the range $[0, 1]$. Hence, values computed by edge label similarity function, κ_{edge}, are also in the range $[0, 1]$. □

4.3.3 Intrinsic Order of Components

In Equation 4.1, the kernel value for two activities represented as TAGs X and Y is computed as the sum of similarity of *label sequences* computed using Equation 4.5. The similarity of the label sequences is computed on a one-to-one basis, i.e. every label sequence of X is matched with exactly one label sequence of Y and vice versa. In order to determine which pair of label sequences from X and Y are compared, the components are assigned an intrinsic order based on: 1. *skeletal structure* and 2. *interestingness.*

4.3.3.1 Skeletal Information

It is possible to track the pose of the human body in video [119, 120]. 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 fact, 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 part 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 $\langle h, rh, lh, rl, ll\rangle$. For two activities represented as X and Y, such that X is defined over u time points and Y is defined over t time points. A *skeletal information based intrinsic order* for X and Y ensures that the similarity of label sequence between heads of the interacting human of X, i.e., $x^{1:u}_{h,h}$ and the label sequence between the heads of the interacting humans in Y, i.e., $y^{1:t}_{h,h}$ is computed. Similarly $x^{1:u}_{h,ll}$ is compared with $y^{1:t}_{h,ll}$; $x^{1:u}_{rh,rh}$ is compared with $y^{1:t}_{rh,rh}$ and so on. The sum of the all the similarity values computed for such one-to-one pairs of label sequences is the kernel value of the corresponding *Temporal Activity Graphs.*

4.3.3.2 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* in short. 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 spatial relational changes with components of the other entity.* The *i-factor of a component* is defined as the sum of the *i-factors* of the label sequences with which the particular component is associated with. Given that, a_i and b_p are components of entities A and B respectively, involved in an activity, the *i-factor of a label sequence* and *i-factor of a component* are defined as follows.

Definition 4.8. *Given a TAG, X, defined over t time points involving extended objects $A = \{a_1, a_2, ...a_m\}$ and $B = \{b_1, b_2, ...b_n\}$, the* ***i-factor of a label sequence****, denoted by $\mathcal{I}_l(x^{1:t}_{i,p}))$ where $ls^{1:t}_{i,p}$ is a label sequence between components a_i and b_p, is computed as follows,*

$$\mathcal{I}_l(ls^{1:t}_{i,p}) = \sum_{u=1}^{t-1} \kappa_{edge}(\varepsilon(a_i, b_p, u), \varepsilon(a_i, b_p, u+1)) \tag{4.8}$$

Here, $x^{1:t}_{i,p}$ is a label sequence between a_i and b_p in the TAG X.

Definition 4.9. *Given a TAG, X, defined over t time points involving extended objects $A = \{a_1, a_2, ...a_m\}$ and $B = \{b_1, b_2, ...b_n\}$, the* ***i-factor of a component*** *a_i, denoted by $\mathcal{I}_c(a_i)$, is defined as,*

$$\mathcal{I}_c(a_i) = \sum_{p=1}^{n} \mathcal{I}_l(x^{1:t}_{i,p}) \tag{4.9}$$

Here, $x_{i,p}^{1:t}$ is a label sequence between a_i and b_p in the TAG X.

In Equation 4.8, it is to be noted that the label sequence $ls_{i,p}^{u:v}$ is $\langle \varepsilon(a_i, b_p, u), \varepsilon(a_i, b_p, u{+}1), ...\varepsilon(a_i, b_p, v)\rangle$ $(u < v)$ for the activity graph X. Since we are working on human interactions, in Equation 4.9, we assume involvement of two entities (A and B with m and n components respectively).

Using Equation 4.9, we compute the *i-factor* for all components of all entities. For each entity the components are ordered in decreasing order of their *i-factors.* We apply this order to compute the kernel value of two activity graphs, X and Y, using Equation 4.1. Thus the label sequence between components with highest *i-factor* of both entities in X is matched with label sequence between components with highest *i-factor* of both entities in Y and so on.

4.3.4 Theoretical Analysis

For a function to be used as a *kernel* function for a SVM, it has to exhibit properties similar to inner product in some implicit feature space. In other words the *kernel* function should be *symmetric* and *positive semi-definite.* In Theorem 4.1, it is proved that the TAG kernel presented in this chapter satisfies these properties.

Theorem 4.1. *The Temporal Activity Graph Kernel, κ, (in Equation 4.1) is symmetric and positive semi-definite.*

Proof. Temporal activity graph kernel κ (Equation 4.1) depends on the label sequence similarity function, κ_{ls} (Equation 4.5) and the edge label similarity function κ_{edge} (Equation 4.7).

By definition the *neighbourhood based similarity* (Definition 4.6) 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 4.2, 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 Equation 4.5 - for exactly similar label sequences the modified edit distance is 0; for all other possible pairs, the value is greater than 0. Using this Equation 4.5 in Equation 4.1 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. Therefore, the *Temporal Activity Graph Kernel* is positive semi definite. □

4.3.5 Illustrative Example

Let us consider the sequence of video frames obtained from a sample *Handshaking* activity in the UT Interaction dataset [3], as shown in Figure 1-1 of Chapter 2. The activity sample involves two human bodies A and B; each human body is an extended object of at most five components. The TAG, X (defined over four time points), for the activity is shown in Figure 4-2. In the figure, components of A are labeled a_p^t, where p refers to the component identifier and t refers to the corresponding time point; similarly components of B are labeled b_p^t. Here, $p \in \{h, rh, lh, rl, ll\}$; 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*.

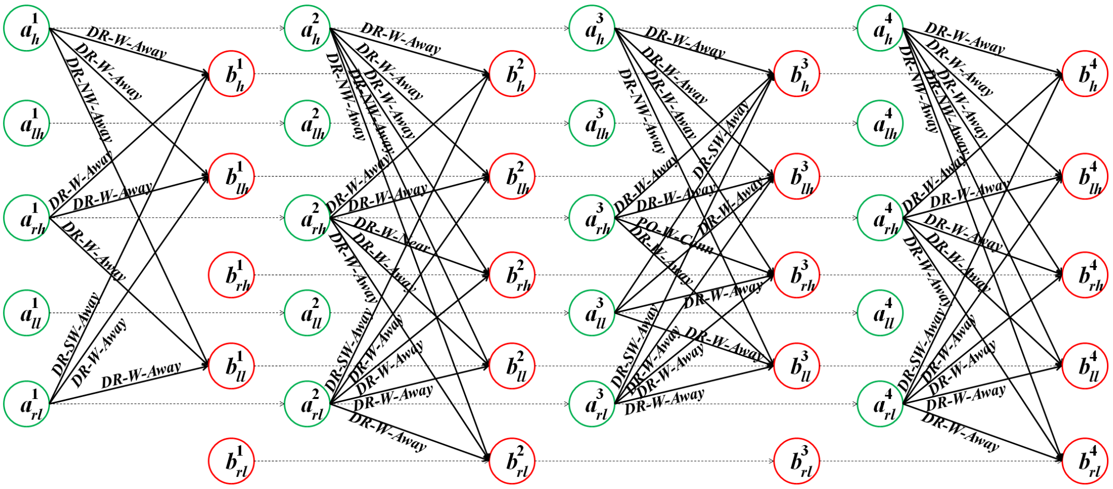

Figure 4-2: TAG for the sequence of four frames in Figure 1-1 that corresponds to *Handshaking* activity from the UT-Interaction Dataset

In the TAG shown in Figure 4-2, the spatial edge labels are three-tuples of topological, directional and distance relations computed using Extended CORE9 as discussed in Chapter 3. 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.

$$
\begin{aligned}
&x^{1:4}_{h,h} = \langle DR - W - Away\rangle\\
&x^{1:4}_{h,lh} = \langle DR - W - Away\rangle\\
&x^{1:4}_{h,rh} = \langle DR - W - Away\rangle\\
&x^{1:4}_{h,ll} = \langle DR - NW - Away\rangle\\
&x^{1:4}_{h,rl} = \langle DR - NW - Away\rangle\\
\\
&x^{1:4}_{rh,h} = \langle DR - W - Away\rangle\\
&x^{1:4}_{rh,lh} = \langle DR - W - Away\rangle\\
&x^{1:4}_{rh,rh} = \langle DR - W - Away\rangle, \langle DR - W - Near\rangle, \langle PO - W - Conn\rangle, \langle DR - W - Away\rangle\\
&x^{1:4}_{rh,ll} = \langle DR - W - Away\rangle\\
&x^{1:4}_{rh,rl} = \langle DR - W - Away\rangle\\
\\
&x^{1:4}_{ll,h} = \langle DR - SW - Away\rangle\\
&x^{1:4}_{ll,lh} = \langle DR - W - Away\rangle\\
&x^{1:4}_{ll,rh} = \langle DR - W - Away\rangle\\
&x^{1:4}_{ll,ll} = \langle DR - NW - Away\rangle\\
&x^{1:4}_{ll,rl} = \langle DR - NW - Away\rangle\\
\\
&x^{1:4}_{rl,h} = \langle DR - SW - Away\rangle\\
&x^{1:4}_{rl,lh} = \langle DR - W - Away\rangle\\
&x^{1:4}_{rl,rh} = \langle DR - W - Away\rangle\\
&x^{1:4}_{rl,ll} = \langle DR - NW - Away\rangle\\
&x^{1:4}_{rl,rl} = \langle DR - NW - Away\rangle
\end{aligned}
$$

Figure 4-3: Set of label sequences describing the TAG shown in Figure 4-2

the sequence of edge labels for $t = 1, 2, 3, 4$. Figure 4-3 shows the label sequences corresponding to the TAG in Figure 4-2. In this example, the label sequence for the components a_{rh} and b_{rh}, denoted as $x^{1:4}_{rh,rh}$, has more relational changes over the duration of the activity. However, the edge labels between the rest of the components do not change over time; the corresponding label-sequences consist of only a single edge label. The component a_{lh} does not appear in any of the time points, therefore all label sequences between components a_{lh} and b_p is $NULL$.

From Definition 4.8, the label sequences with higher relational change will have higher *i-factor*. Using Equation 4.8, $\mathcal{I}_l(x^{1:4}_{rh,rh}) > \mathcal{I}_l(x^{1:4}_{h,rh})$. Consequently from Equation 4.9, $\mathcal{I}_c(a_{rh}) > I_c(a_h)$. Therefore, the intrinsic order of a_{rh} is higher than a_h.

4.4 Experimental Evaluation

The effectiveness of representing activities using TAGs is reflected in the classification results. In order to classify activities represented using TAG, we use the proposed TAG kernel within a SVM classifier. We evaluate the effectiveness of TAG together with TAG kernel against the Extended CORE9 bag-of-words representation. Further the classification accuracies are compared with existing work reported in literature.

4.4.1 Experimental Setup

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. 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. A detailed description of how a video is converted to Temporal Activity Graph representation can be found in Appendix B. The activities represented as Temporal Activity Graphs are then classified using a Support Vector Machine (SVM) based on the kernel function defined in Equation 4.1.

In Equation 4.1, the kernel value for two activities represented as graphs X and Y is computed as the sum of similarity of *label sequence* computed using Equation 4.5. The similarity of the label sequences is computed on a one-to-one basis, i.e. every label sequence of X is matched with exactly one label sequence of Y and vice versa. In order to determine which pair of label sequences from X and Y are compared, the components are assigned an intrinsic order based on: 1. *skeletal structure* and 2. *interestingness.*

4.4.2 Experimental Results

Experiments were performed using 110 videos from the Mind's Eye [1], 50 videos from the UT-Interaction [3] dataset and 282 videos from the SBU Kinect Interaction dataset [122]. For the Mind's Eye dataset we consider 11 activities - *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.* For the UT-Interaction and Mind's Eye dataset, we use keyframes of the videos [2]. Since tracks for these datasets are not available, we manually label the humans and objects involved in each of the keyframes. For the

[1] www.visint.org

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

SBU Kinect Interaction dataset, that consists of eight activities over 282 videos, we use the available skeleton tracks 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. 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 4.1, results for the UT Interaction dataset are given in Table 4.2 and results for the SBU Kinect Interaction dataset in Table 4.3.

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 4.1: 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 4.2: Results for TAG Kernel based SVM Classification on UT Interaction dataset

4.4.3 Discussion

Table 4.4 shows that good classification accuracy is obtained for all the considered datasets, when the skeletal structure is used as the intrinsic order. We have also experimented by considering only topological relation, only directional relation,

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

Method	UTI	ME	SBUKI
ExtCORE9 BoW + KNN [123]	62%	58.18%	40.78%
ExtCORE9 BoW + SVM [123]	74%	55.45%	47.16%
ExtCORE9 BoW + Naive Bayes [123]	74%	55.45%	49.64%
ExtCORE9 BoW + Deep Learning [123]	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 4.4: 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 [123]	74%	55.45%	49.64%
ExtCORE9 BoW + Deep Learning [123]	64%	57.27%	46.45%
Angled CORE9 + LDA [41][a]	-	64.4%	-
BoW + SVM [118]	77%	-	-
BoP + SVM [118]	95%	-	-
Skeleton + Deep LSTM [58]	-	-	86.03%

Table 4.5: Comparison of classification accuracies with other approaches in literature

[a] In [41] only 5 activities are considered; in this thesis 11 activities of the dataset are considered.

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 also works reasonably well. Table 4.4 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 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.

A comparison of classification accuracies obtained using our work and existing literature is shown in Table 4.5. In case of Mind's Eye dataset, our work outperforms Angled CORE9 with a bag-of-words representation [41]. It is to be noted here, that in this work 11 activity classes are considered whereas [41] 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 [124] (0.37). However, we use only 11 activity classes, whereas [124] uses all activity classes in the Mind's Eye dataset.

In Table 4.5, BoP + SVM [118] outperforms our system performance for the UT Interaction dataset. In this work, videos are represented as a bag of spatio-temporal phrases (ST phrases). A spatio-temporal word captures the appearance and movement patterns of a local region. An ST-phrase is a combination of spatio-temporal words in a certain spatial and temporal structure. Whereas using TAG, only a temporal structure is provided to the representation of the video. An ST-phrase includes the order and relative positions of the local regions corresponding to the words. In addition to giving a structured temporal description of the video, the use of ST phrases combines the spatio-temporal descriptions of related local features. This could be one of the reasons why the work reported in [118] outperforms the work reported in the thesis.

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. This is so because the TAG kernel takes into consideration the component-wise relational changes between the extended objects during the label-sequence comparison. It is worthwhile to note that most complex activities involve high component-wise relational changes. However, most activity datasets have a mixture of activities with high and low relational change between the components, which serves as a reason why the TAG kernel does not outperform results in literature.

In Table 4.5, Skeleton + Deep LSTM [58] outperforms the TAG Kernel based work reported in this chapter. It is worthwhile to note 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 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.

4.5 Conclusion

In this chapter, we have presented TAG to represent activities recorded in video. Such a representation is capable of keeping track of the changing spatial relations between entities abstracted as extended objects over the duration of the activity. In order to enable classification of activities using such a graph structure, a TAG kernel is defined. The TAG kernel allows classification of activities represented as TAGs using a SVM.

It is widely accepted that discriminative classifiers such as SVMs provide a good classification of the data. However, they offer little in terms of modelling the underlying structure of the classes [125]. In the next chapter we present a generative learning mechanism, that models the underlying structure of activities represented as TAGs using a *Temporal Activity Graph Grammar*.