Chapter 2

Literature Review

2.1 Human Activity Recognition

Human activity recognition (HAR) has gathered much interest in recent times because of its wide range of applications. Applications of HAR include automated surveillance systems, human-computer interfaces, patient monitoring systems and smart homes among others [18]. Surveillance in public areas like airports, railway stations, parking lots, schools and colleges involve detection of suspicious activities such as terrorism, vandalism, theft, fighting etc. However, continuous monitoring of surveillance videos by a human is difficult and tiring. As such, an automated surveillance systems that learns to identify unusual human activities from video has garnered much interest [19].

HAR is also an important component of automated patient monitoring systems and smart homes. A smart home monitors environmental changes and learns to recognize and predict inhabitant's activities. By doing so, it is able to take appropriate actions and timely decisions to assist inhabitants in performing activities of daily living (ADL) [20, 21]. Such systems allow real time monitoring of patients, children and elderly persons, and further enable them to live independently within their own home environment. Human robot interactions are imminent with an increasing number of mobile robots, particularly domestic robots for cleaning and maintenance of households. HAR is an important aspect for a well-coordinated interaction between humans and robots [22]. Perceptual narratives drawn from HAR in video have also been discussed in the domain of smart-meeting cinematography [23]. Using such narratives to understand the environment allows for intelligent cinematography by automating coordination and control of the cameras.

2.1.1 Categories of Activities

Human activities are known to have an inherent hierarchical and recursive structure [18, 24]. Based on such a view, Aggarwal and Ryoo [18] have categorized human activities as: *gestures, actions, interactions,* and *group activities*. In another recent review presented by Zhang et al [24], human activities have been categorized into three-levels: *action primitives, activities* and *interactions*. The following categorization of human activities is based on the review by Aggarwal and Ryoo [18].

1. *Gestures:* These are the atomic elements of the activities that constitute more complex human activities. Gestures involve a particular limb of the body such as the hands, arms, or upper body part. This is similar to the *action primitive* described in Zhang et al's review. For example, "stretching the left arm" and "raising the right leg" are some actions that belong to this category.

2. *Actions:* These are the human-activities that may be composed of more than one gesture but involve only a single human. This category is termed as *activities* by Zhang et al. For example, "walking", "waving", "pointing" belong to this category.

3. *Interactions:* Interactions are complex activities that involve two humans or a combination of a human and an object. For example, activities like "handshaking", "kicking" belong to this category. Aggarwal and Ryoo [18] further categorize interactions between two humans as *human-human interaction* and interactions between a human and an object as *human-object interaction.*

4. *Group Activities:* Activities that involve groups of more than two humans and objects are termed group activities. This category is not separately recognized by Zhang et al; however, they describe *interactions* as activities involving more than one person or object. This allows *group activities* to be clubbed with *interaction* in Zhang et al's categorization. Typical examples in this category are "group of people having a meeting", "a group of people playing football" etc.

Along similar lines, Vrigkas et al [2] recognize further higher level categories such as *events* and *behaviors*. According to their categorization, *behaviors* refer to activities that are associated with the emotions, personality, and psychological

state of the individual. *Events* are high-level activities that describe social actions between individuals and indicate the intention or the social role of a person. Elsewhere, *events* have also been defined as *any* change in the spatio-temporal state of a physical object during an interval of time [25]. Considering such a varied range of terminology used in literature, we restrict ourselves to using the terminology defined by Aggarwal and Ryoo [18]. Further, this thesis is centred around *human interactions*.

2.1.2 Levels and Stages of HAR

In a review presented by Ke et al [26], the study of HAR systems is separated into three levels as shown in Figure 2-1. In the first level, study is focused on the low-level core technologies of an HAR system. At this core-technology level, there are three basic stages of HAR - (a) *Detection and tracking of objects in the video*, (b) *Extraction of features for representation of activities*, and (c) *Reasoning mechanisms for classification of activities*. The entities involved in the activity are detected and tracked within the video. Based on the tracking data, interesting characteristics of the involved entities are extracted. The extracted features from the video are used for an appropriate representation of the activity. Subsequently, classification algorithms are applied on the activities for HAR.

In the second level, mid-level human activity recognition systems are discussed. Based on the classification of human activities in Section 2.1.1, there are four basic types of HAR systems - (a) *gesture* recognition, (b) *action* recognition, (c) *interaction* recognition, and (d) *group activity* recognition. Finally, in the third level, discussion surrounds high level application of HAR systems.

It is seen that over the years representation approaches have made transition from global representations to local representations, and most recently to depth-based representations [24]. Similarly, the learning mechanisms used within HAR keep evolving. It has been noted that lots of techniques that were not specifically designed for HAR have received much attention. For example, Dynamic Time Warping, Hidden Markov Models and more recently deep learning techniques are popularly used within HAR [24]. Object recognition and tracking techniques are related to the HAR scope and has seen considerable progress over the years as well.

A generalized framework for multilevel human activity analysis framework has been recently proposed in [27]. The framework is a combination various modules as follows -

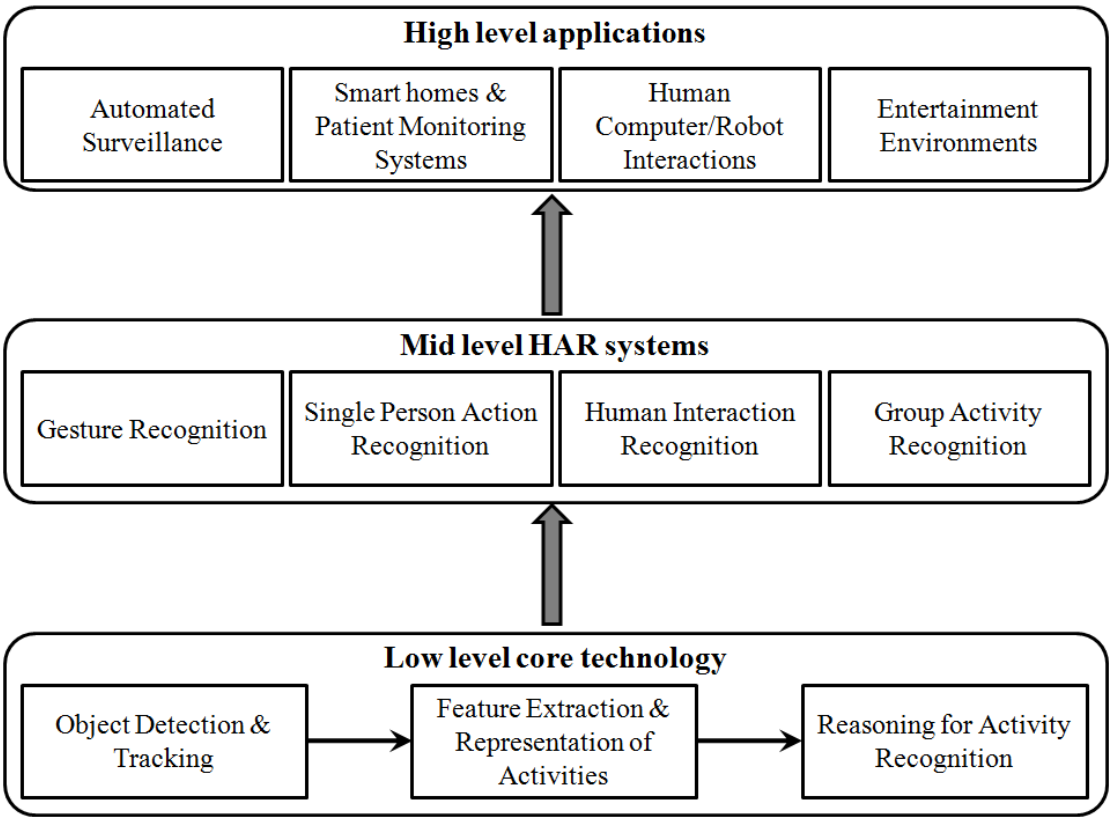


Figure 2-1: Overview of an HAR system[26]

(1) low level feature extraction,

(2) Laban movement analysis parameters[1],

(3) human movement estimation,

(4) interpersonal behaviors estimation,

(5) activity estimation, and

(6) social role estimation

It is worth noting that the framework in [27] can be seen as a more fine-grained view of the pipeline shown in Figure 2-1. The modules (1) and (2) are easily recognized as the *feature extraction and representation* phase of low-level core technologies. Modules (3), (4), (5), and (6) may fall within the *reasoning for activity recognition* phase. The framework additionally shows how the various *reasoning modules* interact within a *mid-level* view of HAR systems.

[1]Laban movement parameters were proposed for describing, annotating and interpreting human movement in the field of choreography [28]

In this thesis, the focus will be on the low-level core technologies for *feature extraction*, activity *representation* and *reasoning* mechanisms for activity recognition. The object detection and tracking data are assumed to be available via existing technologies discussed in literature [29, 30, 31].

2.1.3 Representation Schema

Researchers have used a multitude of features for describing the events occurring within a video. Most vision based HAR systems use global features such as silhouettes [32], contours [33], optical flow [34]. Such global descriptors of activities are known to be extremely sensitive to noise, occlusions and viewpoint variations [7]. Grid based representations that divide the images into fixed spatial and temporal grids are seen to partially solve the problems in global representations. Descriptors such as histogram of oriented gradients (HOG) [35], grid-based optical flow [36], silhouette and flow grids [37] have also been used. Other global representation schemes involve use of space-time volumes obtained by stacking sequences of silhouettes to obtain a 3D volume [38].

It has often been argued that humans do not perceive images in terms of pixel information [39]. Following this argument, some worked on building a 3D representation of components from the 2D images [40]. Many researchers have also focused on representation of events in terms of objects and their qualitative spatio-temporal relations [5, 11]. Within qualitative frameworks for representation of activities, various aspects of space, such as topology and direction, have been considered. CORE9 is a compact and comprehensive representation framework that encodes topological, directional, size, distance and motion information of a pair of objects abstracted using axis-aligned rectangles [13].

Within qualitative frameworks, humans are usually abstracted using bounding boxes [5, 13, 41]. However, for human activities, using a single bounding box for the whole body abstracts away a lot of interaction details. A part-based model of the human body [29, 30, 31] is seen to help solve the problem to some extent [15]. This is comparable to how grid-based representations are better descriptors than global representations, as discussed earlier. However, simply increasing the granularity does not take into account part-whole relations. Moreover, increasing granularity in a qualitative framework may lead to an explosion of relations many of which may be redundant. To alleviate this to some extent, in this thesis we discuss *extended object* based abstraction and a representation that uses qualitative spatial relations between extended objects.

Qualitative spatial relations describe the configuration of objects in space at an instant of time. However, activities are spatio-temporal in nature and to encode temporal information of activities various techniques have been adopted. A sequence of spatial relations have been used to describe an activity [42]. Time is also expressed using qualitative temporal relations [43] in hierarchical graph representation [11] as well as first-order logic formulations [5]. Temporal nature of activities have also been discussed by keeping track of objects trajectories and the qualitative relations between these trajectories [44]. Graph based representations have also been often for representation of the spatio-temporal structure of activities [11, 45, 46].

2.1.4 Reasoning Mechanism

Recognition of events and activities within a video is generally done by generating models for event classes [5]. There exist works that focus on classification without considering the temporal structure of the event. Some such works have used classifiers, like k-Nearest Neighbours, to obtain action class descriptions [32, 33]; a prototype of the classes is obtained as the mean of all examples within the classes. Discriminative classifiers such as Support Vector Machines (SVM) [47] have also been used. Many researchers have also worked on learning models that focus on the temporal structure of the events. Hidden Markov Models (HMM) have been used in many of the works such that states correspond to different stages of the events [48, 49]; Conditional Random Fields have also been used [50, 51]. Based on various spatio-temporal graph representations for activities different learning techniques have been used such as - *graph based relational learning* [11], *graph-kernel based classification* [46] among others. Grammar-based approaches have also been reported in the literature [52, 53, 54, 55]

Of late, much research has focused on data driven algorithms [56, 57, 58]. 3D Convolutional Neural Networks have been used for learning spatio-temporal features [56]; Recurrent Neural Network based HAR has been shown to achieve high accuracy [57]; Recurrent Neural Networks with Long Short-Term Memory have also been used to learn feature representations and model long-term temporal dependencies for HAR [58]. Such techniques focus on the pixel-information obtained from very large datasets to achieve high accuracy; they fail to take into account high-level contextual information contained in the video [10]. High-level contextual information could give interesting insights for learning interaction models from relatively small datasets.

2.1.5 Knowledge Representation & Reasoning for HAR

Knowledge Representation and Reasoning (KR & R) is concerned with how symbolic *knowledge*, instead of quantitative information, can be used in an automated system for reasoning. The rising popularity of KR & R methods in the area of video understanding is because of the recognition that more conceptual and generic models can be learned [13]. Qualitative abstractions of space-time are often used in such systems for representation of knowledge pertaining to the activities [11, 41, 42, 59].

Several qualitative representation systems revolve around reasoning about moving objects in space [60, 61]. This allows for a natural transition to using qualitative reasoning for understanding and reasoning about objects moving in space over time, i.e. events within a video. Reasoning about occlusion relations have been used to detect and eventually recognize multi-object events within a video [62]. Constraint based logical reasoning for qualitative representations have been used to improve upon tracking of moving objects within video [63]. Simple events involving inanimate objects, such as vehicles in a traffic surveillance systems, are often seen within such discussions [62, 64].

The learning mechanism used in the system is usually tied to the underlying representation of activities. When a knowledge-based representation is used, logic based reasoning techniques can be utilized for learning complex human activity models. Relational learning techniques such as Inductive Logic Programming have been used when video events are represented as first-order logic formulae [59]. Graph based relational learning has been explored where events are represented using *interaction graphs* and *activity graphs* [11]. Grammar based approaches have also been discussed for recognition of activities wherein activities are described using first-order logic predicates [18]. Although the grammar rules are not learned or induced in [18], it has been noted that grammar based approaches are most suited to capture the hierarchical and recursive structure of human activities.

Declarative reasoning has also been used for a high-level interpretation of ongoing activities within a video [23, 65]. It has also been used for semantic interpretation of sensor data, such as those obtained from object tracking, eye tracking data, movement trajectories in a video [65]. Constraint Logic Programming is then used for a high-level explanation of observed events and answering questions about perceived interactions in the video. Such perceptual narratives of activities obtained using knowledge-based reasoning have been discussed within a smart meeting scenario for automated cinematography [23].

There are also examples in literature where a knowledge-based representation of activities have been used together with non-logic based learning mechanisms. Such systems take advantage of the high-level intuitive representation of activities and the simplicity of non-logical learning algorithms. Popular similarity-based learning techniques such as those employing Support Vector Machines have been used together with qualitative spatio-temporal representation of activities [42]. Probabilistic learning algorithms such as Latent Dirichlet Allocation has also been used for unsupervised learning of activities [41, 44]. The qualitative knowledge based description of activities are treated as a bag-of-words description in such approaches. Recent works have also incorporated deep learning techniques with a qualitative description of activities [66]. Therein, qualitative abstractions of direction, distance, and trajectory relations between the interacting objects are used to describe an activity. Such qualitative descriptions are used with Long Short Term Memory (LSTM) for learning temporal sequencing of events and a Multi-layer Perceptron for learning event models.

KR & R techniques are gaining popularity in the field of video understanding. On one hand, success with KR & R techniques for simple events involving moving objects [62, 64] has lead to the use of such techniques for modelling more complex human activities [23, 41]. On the other hand, efforts are being made for taking advantage of the rich descriptions for a more intuitive understanding of complex activity sequences [65]. Furthermore, there is emerging a trend for using symbolic descriptions within non-symbolic learning techniques [41, 42, 44, 66]. In this thesis, we present a graph based representation of activities. Herein, symbolic knowledge pertaining to spatial information of the entities are embedded in the form of qualitative relations. For learning within such a representation, we discuss non-symbolic classification techniques in Chapters 3 and 4. In Chapter 3, popular off-the-shelf classification techniques are utilized with a qualitative relational bag-of-words description of activities. In Chapter 4, symbolic knowledge is incorporated as qualitative relations within a graph-based representation for activities. Consequently a more specialized kernel-based SVM classification is discussed. A grammar based learning mechanism is presented in Chapter 5 that utilizes symbolic reasoning techniques for learning complex activity models.

2.2 Qualitative Spatial Representation

Qualitative Reasoning within KR & R is concerned with the qualitative abstractions of the physical world in order to capture non-metrical common-sense knowl-

edge. Given appropriate reasoning techniques, the behaviour of physical systems can be explained without having to fall back on intractable or unavailable quantitative models. Qualitative Spatio Temporal Reasoning (QSTR) provides formalisms for capturing common-sense spatial and temporal knowledge [67]. These formalisms usually rely on a relational description of space-time events rather than actual quantitative details of object locations and durations of the happenings. Researchers often use such formalisms for a more intuitive description of video activities [11, 59, 68].

Within QSTR, formalisms have been defined for qualitative representation and reasoning for various aspects of space [69]. Topology [70], direction [71], motion [60], order [43], distance and size [72] are few aspects of space that have often been discussed in literature. Such a formalism, often termed a *qualitative calculus*, is built upon a set of *jointly exhaustive pairwise disjoint* (JEPD) relations called the *base* relations [73]. The base relations are a finite set of relations that partition the entire set of possible pairs of elements for a non-empty universe. Between any pair of elements from the universe, exactly one of the base relations is possible.

In this thesis, we discuss binary qualitative relations between 2D rectangle regions for a relational representation of the spatial configuration of objects during an activity. Further, this thesis is mostly focused on the computation of qualitative relations for the aspects of topology, direction, and distance.

2.2.1 Topology

Topology deals with relations unaffected by change of shape or size of objects. A notable framework for expressing qualitative topological relations between a pair of regions, having the same dimension, is the Region Connection Calculus [70]. In this approach relations are defined based on a *connection* primitive, where two regions are said to be *connected* if they share at least one common point. Two subsets of *jointly exhaustive and pairwise disjoint* relations, viz. RCC5 and RCC8, are widely used by researchers. The Region Connection Calculus (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). Figure 2-2 shows the spatial configuration of two spherical objects with different RCC8 relations. In RCC5, relations DC and EC are abstracted by single relation Disjoint(DR), relations TPP and nTPP are abstracted as Proper Part (PP), and relations TPPi

and nTPPi are abstracted as Proper Part Inverse (PPi).

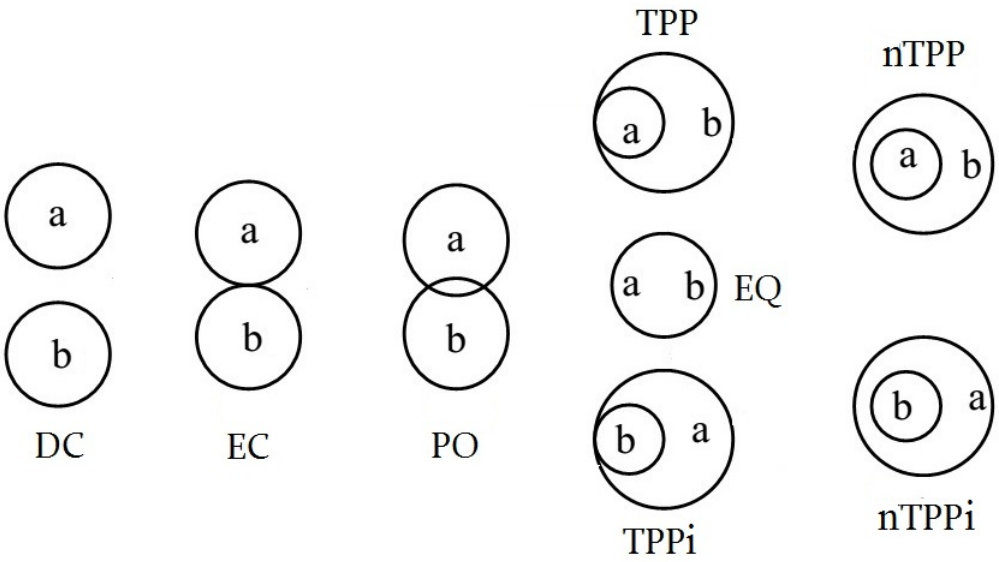


Figure 2-2: The JEPD relations of RCC8 [70]

2.2.2 Direction

Qualitative directional relations have also been extensively discussed in literature [71, 74]. Qualitative directional information may either assume a global frame of reference [71] or a relative frame of reference using a reference object [74]. In this thesis, we shall focus on cardinal direction relations that are based on a global frame of reference. Cardinal direction relations are one of the eight cardinal directions: North (N), NorthEast (NE), East (E), SouthEast (SE), South (S), SouthWest (SW), West (W), NorthWest (NW) [71]. Cardinal direction relation relations have also been discussed with respect to objects with multiple components [75]. For multi-component objects, a combination of cardinal direction relations are used. Figure 2-3 shows how cardinal direction relations are defined for simple objects and objects with multiple components.

2.2.3 Distance

Researchers have also discussed distance and size relations in a qualitative framework. Qualitative size relations have been defined within a mereological framework of part-whole relations, that is extended by primitive relations *same-size-as* and *roughly-the-same-size-as* [72].

The qualitative distance relations are defined using a combination of *connection* primitive, *sphere* primitive and the qualitative size relations. The sphere primitive

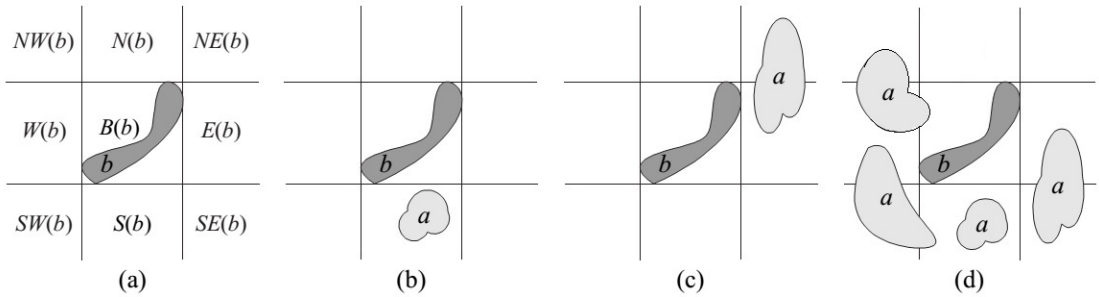


Figure 2-3: (a) Cardinal directions of b (b) a S b (c) a NE:E b (d) a B:S:SW:W:NW:E:SE b [75]

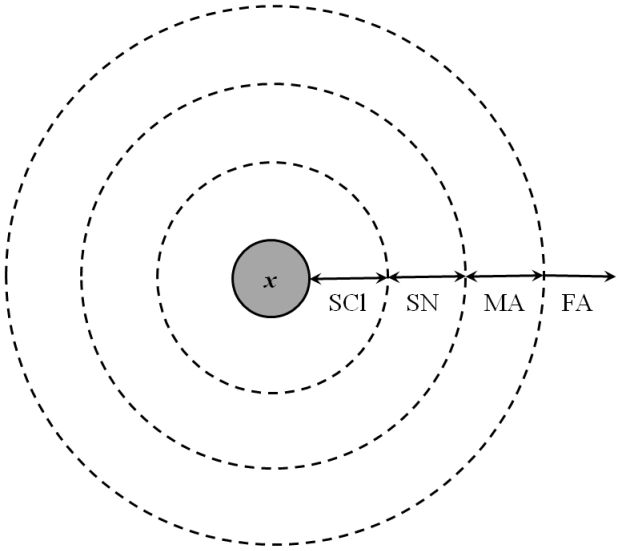


Figure 2-4: Qualitative distance for a spherical object x [72]

defines a region to be a sphere. Within such a framework, qualitative distance relations Close (Cl), Strictly Close (SCl), Near (N), Strictly Near (SN), Away (A), Far Away (FA), Moderately Away (MA) are defined for two disconnected sphere regions. Figure 2-4 shows the distance ranges corresponding to the base relations SCl, SN, MA, and FA are shown. As shown in the figure, given a spherical object x, another object y is said to be SCl if it lies strictly within the innermost circle and so on. It is to be noted that if two objects are topologically *connected* then the distance between them is zero and the qualitative distance is said to be Connected (C).

2.2.4 Extended Objects

In the frameworks discussed above, objects are assumed to be simple, connected regions. In literature, extended objects, which can be seen as a set of disconnected regions, have also been discussed. Extended objects have been referred to as *com-*

posite regions [16] and topological relations between composite regions are defined at two levels - the *coarse* level and *detailed* level. At the coarse level, a single general relation is used to express the topological relation between the composite regions as a whole. At the detailed level the relations at the component level are considered within a *reduced topological matrix*; here only relations between a component of one region and a component of the other region are considered. Elsewhere, topological relations between complex objects, which are seen as a set of discrete points, disjoint lines, or disjoint regions is discussed [17]. A concept of *predicate clustering* is proposed to express component relations as a single general relation, exclusively resting on the emptiness and non-emptiness of component intersections. Topological relations between regions with holes have been discussed in [76]; wherein, in addition to the relation between the whole regions, the topological relations between the holes of one region and the the holes of the other region are considered. It is assumed that only the relations between holes of different regions are necessary, as the relation between holes of the same region can be implicitly understood to be *disjoint*. Direction relations between extended objects have also been discussed [75]. The space around an object is divided into nine tiles corresponding to the eight cardinal directions surrounding it and one central box area containing the object. The relation of a union of regions to another region is expressed as a sequence of cardinal directions corresponding to each of the tiles of one region that the other region may overlap (see Figure 2-3d). To the best of our knowledge qualitative distance relations for extended objects have not been discussed in literature.

2.3 CORE9 and its variants

CORE9 was proposed as a comprehensive rectangle representation that allows an integrated representation of several interesting spatial information between two rectangles, viz. topology, direction, size, distance, and motion [13]. Their focus is on presenting a compact representation for one-piece, rectangular, axis-aligned regions, which is a common abstraction of objects in video analysis rather than a more precise shape representation. By extending the boundaries of the pair of rectangle objects, the *region of interest* is divided into nine *cores* as shown in Figure 2-5. By maintaining the *state* information for these nine *cores* it is possible to infer the topological and directional information of a pair of objects. The state information of $core_{i,j}(A, B)$ is $state_{i,j}(A, B)$ and can have the following values:

(a) AB if the core is a part of $A \cap B$

(b) A if the core is a part of $A - B$

(c) B if the core is a part of $B - A$

(d) $\square$ if the core is not a part of A or B

(e) ϕ if the core is only a line segment or point

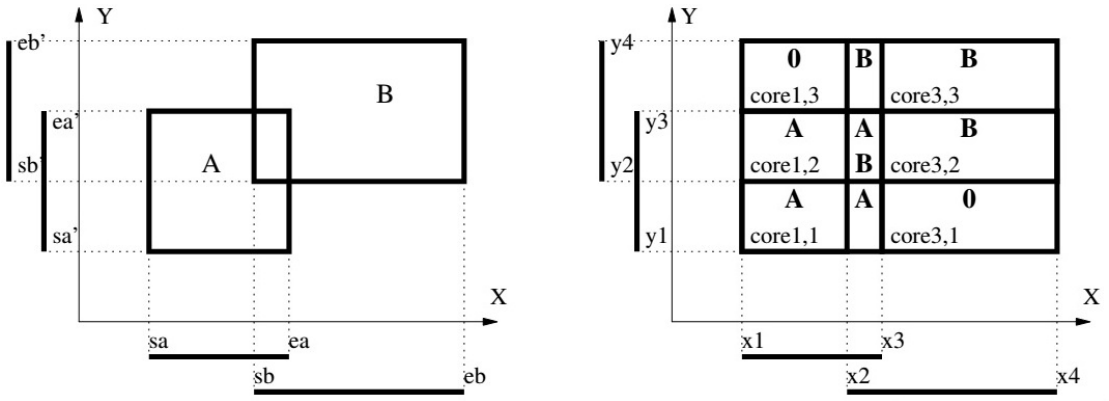


Figure 2-5: The RoI and 9 cores of CORE9 [13].

Further, the cores can be ranked based on their coverage area. The distance, size and motion information, can be inferred from the *ranking* information of the nine *cores*. The relative size and distance information can be obtained straightforwardly by comparing which core is larger than which other core. The relative motion information can be obtained by comparing the size of a core at one time point to its size at the next time point.

2.3.1 Reasoning within CORE9

Given objects A and B, the state of objects A and B, is the 9-tuple represented through [$state_{1,1}(A,B)$, $state_{1,2}(A,B)$, $state_{1,3}(A,B)$, $state_{2,1}(A,B)$, $state_{2,2}(A,B)$, $state_{2,3}(A,B)$, $state_{3,1}(A,B)$, $state_{3,2}(A,B)$, $state_{3,3}(A,B)$]. The state of objects A and B in Figure 2-5, written in a matrix form for clarity, is as follows.

$$state(A,B) = \begin{bmatrix} A & A & \square \\ A & AB & B \\ \square & B & B \end{bmatrix}$$

From this *state information*, it is possible to infer that the RCC-8 relation between A and B is PO, because there is at least one core that is part of both A and B.

Based on the position of the cores occupied by A and B, the CDC relation can be inferred to be $A : SW : B$. Further, the state information is seen to have a one to one correspondence to the topological-directional relation, given as a Rectangle Algebra (RA) relation [2]. Inferring the topological relation as an RCC-8 relation from a given RA relation is straightforward for convex, axis-aligned rectangles.

Furthermore, based on the coverage area of the cores, the ranking information for A and B can be computed as follows.

$$rank(A, B) = \begin{bmatrix} 6 & 3 & 9 \\ 4 & 1 & 7 \\ 5 & 2 & 8 \end{bmatrix}$$

Given the rank and state information, the total size of the cores occupied by object A is smaller than the total size of the cores occupied by B. Therefore size of A is inferred to be smaller than size of B. On the other hand, the distance between A and B is zero since they are topologically connected. The motion information can be inferred by comparing rank and state information between A and B in the next instant of time.

2.3.2 Variants of CORE9

It has been noted that objects within a video are often represented using their minimum bounding rectangles (MBR) [13]. Based on such an observation, CORE9 was designed for a pair of axis-aligned rectangular objects, allowing for several representational efficiencies. However, it has been argued that relations obtained for a pair of objects, abstracted using a single axis-aligned rectangle, are often inaccurate [41]. As shown in Figure 2-6, the topological relation between the two regions is computed to be *partially overlapping* using CORE9. The figure shows that using *oriented* rectangles for abstracting the regions, the regions are accurately found to be topologically *disjoint*. Angled CORE9 was presented for dealing with such oriented rectangles [41]. The emphasis in Angled CORE9 was on computing appropriate orientations of the rectangles that could be used for computing accurate qualitative relations. They have utilized the Maximum Margin technique, popularly used with Support Vector Machines, for identifying the hyperplane that separates the regions. They go on to show that this hyperplane also provides the best angle of orientation for the rectangles that bound the regions.

[2] A depiction of the states and their correspondence with the RA relations can be found in www.comp.leeds.ac.uk/qsr/cores

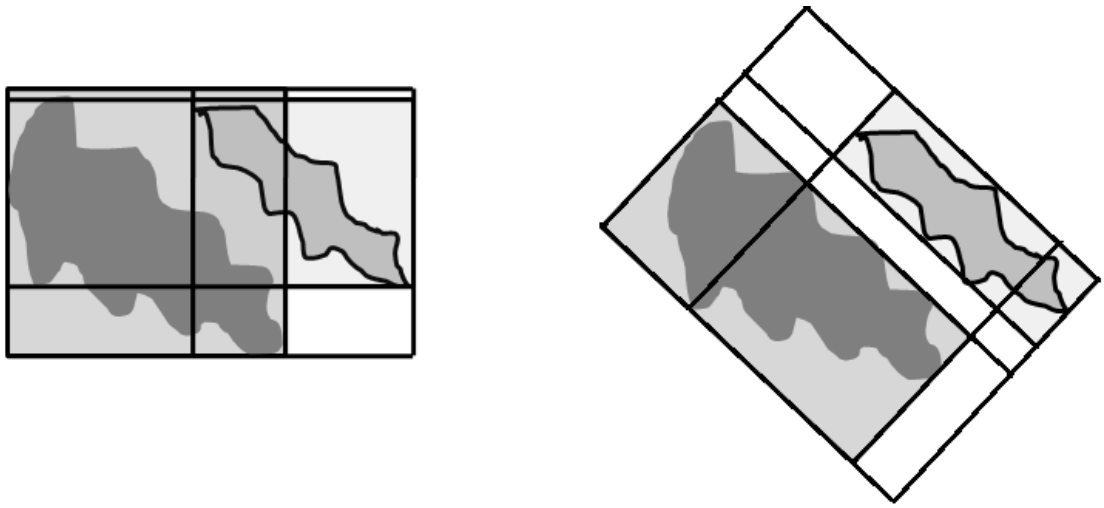


Figure 2-6: Inaccurate relations are computed if the objects are abstracted using a single axis-aligned rectangle

Researchers have also discussed an extension of Angled CORE9 for 3D oriented rectangles wherein the entire video is seen as a spatio-temporal volume [44]. Therein, an approach is presented for adapting Angled CORE9 to be used with spatio-temporal volumes. The video is segmented spatio-temporally to obtain the spatio-temporal volumes of the interacting objects from the video. The activities are described based on the spatial and temporal aspects computed for the spatio-temporal volumes.

Both variants discussed above define techniques that allow CORE9 to be used for better abstractions of the entities involved in the activities. This is done by using a single oriented bounding rectangle for abstracting an entity in the video. This suggests that a proper abstraction of the interacting entities within the HAR system affects a large number of factors, including efficiency and effectiveness. For inanimate objects that do not have individual movable parts, an abstraction using a single bounding box may suffice. However, for human bodies that have movable parts, such as hands and feet, an *extended object* based abstraction that uses multiple rectangles for each individual movable part seems to be more appropriate. In this thesis, we define *extended objects* as a set of components, such that each component is approximated by an *axis-aligned minimum bounding rectangle*. Therefore, in Chapter 3 we present Extended CORE9, a mechanism for obtaining the spatial relations between the interacting entities. To this extent, we use geometric reasoning to reduce the amount of computation, thereby enhancing efficiency.

2.4 Qualitative Reasoning

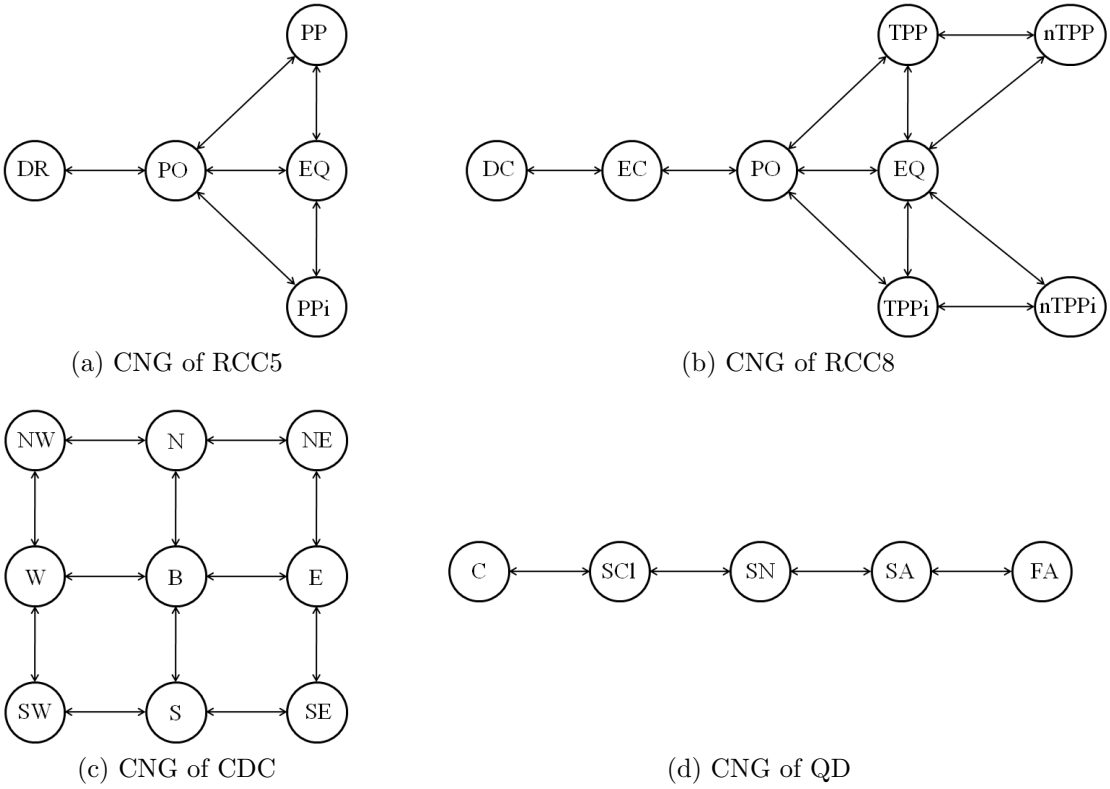
Within QSTR, majority of the research is focused on the representation of typically infinite spatial and temporal domains using a finite set of symbols [67]. Nonetheless, constraint-based reasoning techniques have been popularly used for manipulation of such qualitative representations. The general constraint based reasoning techniques of QSTR have also empowered other forms of similarity-based or logic-based reasoning [69]. Constraint-based reasoning usually surround the idea of *compositions* of relations of a qualitative calculus [67]. Given three entities in a spatial/temporal domain, a, b, c, if the relation between a and c is known to be R_1 and the relation between b and c is R_2, then compositions allow inference of the relation between a and c. Such compositional inferences are independent of the domain elements and can be seen as *constraints* on the relation between a and c. In a recent review by Dylla et al [69] the following forms of constraint-based reasoning are recognized: (a) *Constraint network generation*, (b) *Consistency checking*, (c) *Model generation*, and (d) *Equivalence transformation*

Other forms of reasoning within QSTR involve *conceptual neighbourhood graphs*. Conceptual neighbourhood graphs are particularly useful for reasoning about spatial entities that change over time. Subsumption lattices are logical reasoning tools that are also used with qualitative representation [77].

2.4.1 Conceptual Neighbourhood Graph

A conceptual neighbourhood graph (CNG) for a qualitative calculus is a directed graph with nodes corresponding to a single base relation. The edges in a CNG are defined based on the assumption that change within the spatio-temporal domain is *continuous* [67]. 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. 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 [78]. Continuous transformation of entities can mean either continuously moving, shortening or lengthening the entities. CNGs are important tools for reasoning in a qualitative framework. In Section 4.3.2 of Chapter 4, we use CNG to compute the similarity between a sequence of relations between a pair of interacting entities.

Figure 2-7 shows the CNGs for RCC5, RCC8, Cardinal Direction relation and Qualitative Distance relations. For example, in Figure 2-7(a), an edge from



(a) CNG of RCC5

(b) CNG of RCC8

(c) CNG of CDC

(d) CNG of QD

Figure 2-7: Conceptual Neighbourhood Graphs (CNG) of RCC5 and RCC8 [70], CDC [79], and QD [72]

relation PO to EQ indicates if the relation between two entities a and b at time t is PO then it is possible that the relation between a and b at time $t+1$ is EQ. Likewise, since there is no edge from DR to EQ, it indicates that it is impossible for the relation between a and b to continuously change from DR to EQ. In order, for the relation a and b to change from DR to EQ, it has to change from DR to PO and then to EQ. In general, if there is an edge in the CNG from R_1 to R_2, then there is also an edge from R_2 to R_1. The relations R_1 and R_2 are said to be *conceptual neighbours*.

2.4.2 Subsumption Lattice

The base relations of a qualitative calculus, together with other more general relations defined within the calculus can be arranged in a subsumption hierarchy or subsumption lattice. The subsumption lattice is constructed using the partial order *subsumption* relation. A relation R_1 is said to subsume another relation R_2 if R_1 holds whenever R_2 holds between a pair of entities. The relation R_1 is a more general relation then R_2 and appears higher up in the lattice. The most general relations are connected together to the *top* ($\top$) relation. The most specific relations

in the hierarchy are connected together to the *bottom* ($\perp$) relation. Figure 2-8 shows the subsumption lattice of RCC [70]. In addition to the base relations of RCC8 and RCC5, the subsumption lattice of RCC includes other general relations such as *part* (P), *part inverse* (Pi), *overlap* (O) and *connection* (C).

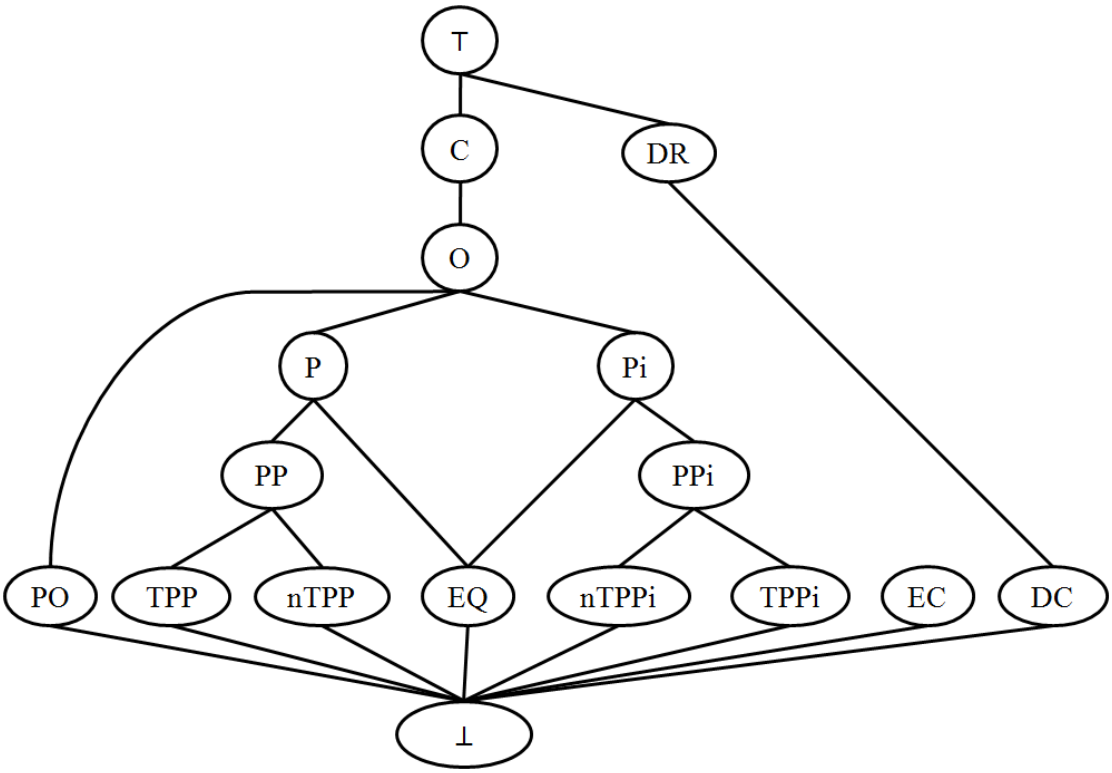


Figure 2-8: Subsumption Hierarchy of RCC [70]

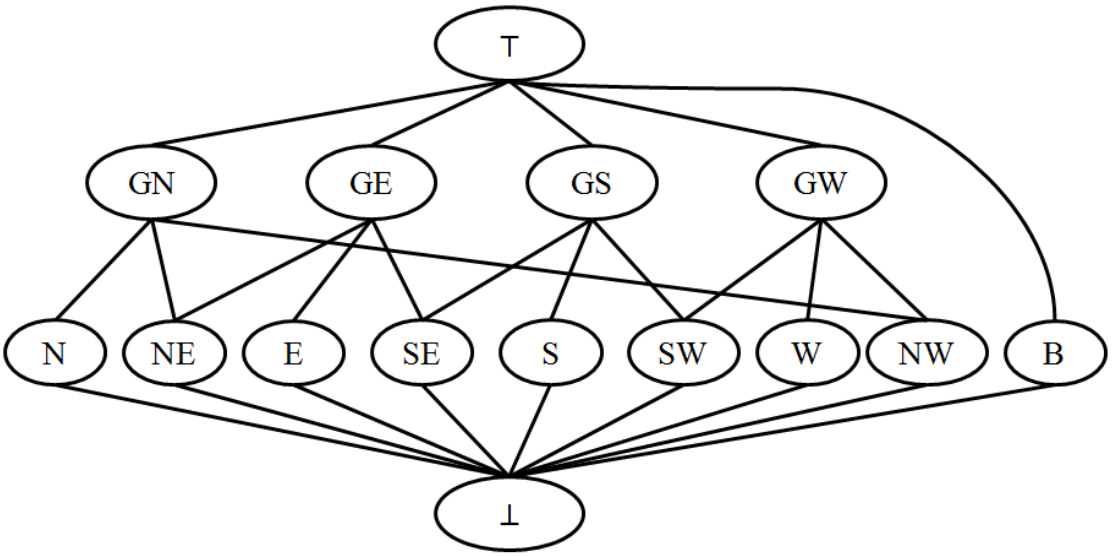


Figure 2-9: Subsumption Hierarchy of Cardinal Direction Relations

Similarly, the subsumption lattice of the Cardinal Direction Relations can be constructed as shown in Figure 2-9. Here, in addition to the base relations, the

general north (GN), *general south* (GS), *general east* (GE), and *general west* (GW) relations are used in the lattice.

2.5 Geometric Reasoning

Geometric Reasoning is defined as "*the process of defining and deducing properties of a geometric entity using intrinsic properties of entity, its relationships with other geometric entities, and the rules of inference that bind such properties together in geometric space*" [80]. However, this has been discussed in literature from two different angles. The first approach involves the traditional use of algebraic expressions for representation of geometric information [81]. This enables generic solutions to geometric problems such as finding intersection of geometric entities etc. In another approach, the properties of the geometric entities within such a system is captured using high-level constructs such as first-order predicates [80]. The reasoning of these properties is supported by the deduction within first order reasoning. In this thesis, we use the latter interpretation of geometric reasoning. A geometric reasoning framework is discussed in Chapter 3 for efficient computation of qualitative spatial relations between entities. Qualitative spatial relations can be seen as a first-order representation of the properties of the geometric entities within a video frame.

2.5.1 Qualitative-Geometric Reasoning

For most cases, qualitative and geometric reasoning go together. Analog geometric representations provide the basis from which the qualitative spatial descriptions are obtained [82]. Qualitative models of space provide a more intuitive description of spatial and temporal information. Such systems allow for a common-sense reasoning about the spatio-temporal domain. However, for a computer, processing of quantitative geometric information is more straightforward. As such, models have been proposed for a hybrid mechanism that interleaves qualitative reasoning with computational geometry methods for reasoning [83]. The system proposed therein takes as input both qualitative and quantitative descriptions for a richer and more efficient reasoning about spatial data. Qualitative reasoning has also been integrated with numerical computation for military planning and alternative battle plans [84].

Qualitative spatial and temporal relations have also been used together with quantitative spatial features for recognition of activities of daily living [42]. In this

work, qualitative spatial features are the topological RCC [70] relations between objects and qualitative temporal relations are the Allen's Interval Algebra [43] relations. Quantitative spatial features include Euclidean distances between objects and relative direction of motion. Feature selection techniques are applied on the entire set of qualitative and quantitative features before learning activity models using a Support Vector Machine. Although this technique does not explicitly combine qualitative reasoning with geometric reasoning, the representations are combined for a richer description.

2.6 Graph Representation

Graphs are a popular mechanism for representation of relations between pairs of objects. The set of *vertices* may be used to represent the objects; the relation between pairs of objects, are represented as the set of *edges* between vertices. Graphs have found a large number of applications within computer science, including computer networks, world wide web, social networks, computer vision, and bioinformatics among others.

A graph consists of a set of vertices or nodes[3] (V) and a set of edges ($E \subseteq V \times V$). Several variations on graphs have been used and implemented. In an *undirected* graph, there are no directions on the edges in the graphs. That is, if there is an edge $(a, b) \in E$ then it implies that $(b, a) \in E$. In a *directed* graph, the edges have directions; if there is an edge $(a, b) \in E$ then it is not necessarily true that (b, a) is an edge in the graph. Figures 2-10a and 2-10b show examples of undirected and directed graphs respectively.

In another variation of graphs, called *weighted* graphs, numerical weights are assigned to the edges. It is also possible to assign weights to the nodes of the graphs. Figure 2-10c is an example of weighted directed graph. A variation of weighted graphs are *attribute* graphs, where instead of numerical weights, the edges are labeled with non-numerical *attributes*. Attributes are descriptors that can be in the form of string, a function, etc. Figure 2-10c is an example of attribute graph where the attributes are the topological (RCC) relations between corresponding objects. Within QSTR, such directed attribute graphs where attributes are qualitative relations between entities are termed as *qualitative constraint networks* (see Section 2.4). *Hypergraphs* are a form of graphs where edges maybe defined between more than a pair of vertices. The hypergraph in Figure 2-10e has

[3]The terms nodes and vertices are used interchangeably within this thesis.

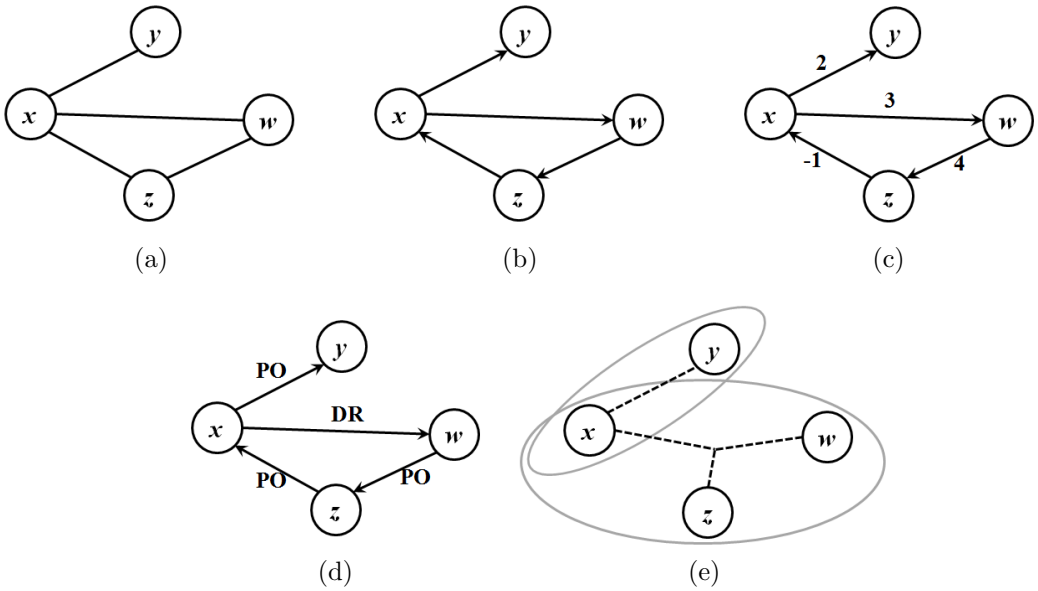


Figure 2-10: (a) Undirected Graph (b) Directed Graph (c) Weighted Directed Graphs (d) Attribute Graph (e) Hypergraph

two *hyperedges* - $\{x, y\}$ and $\{x, w, z\}$.

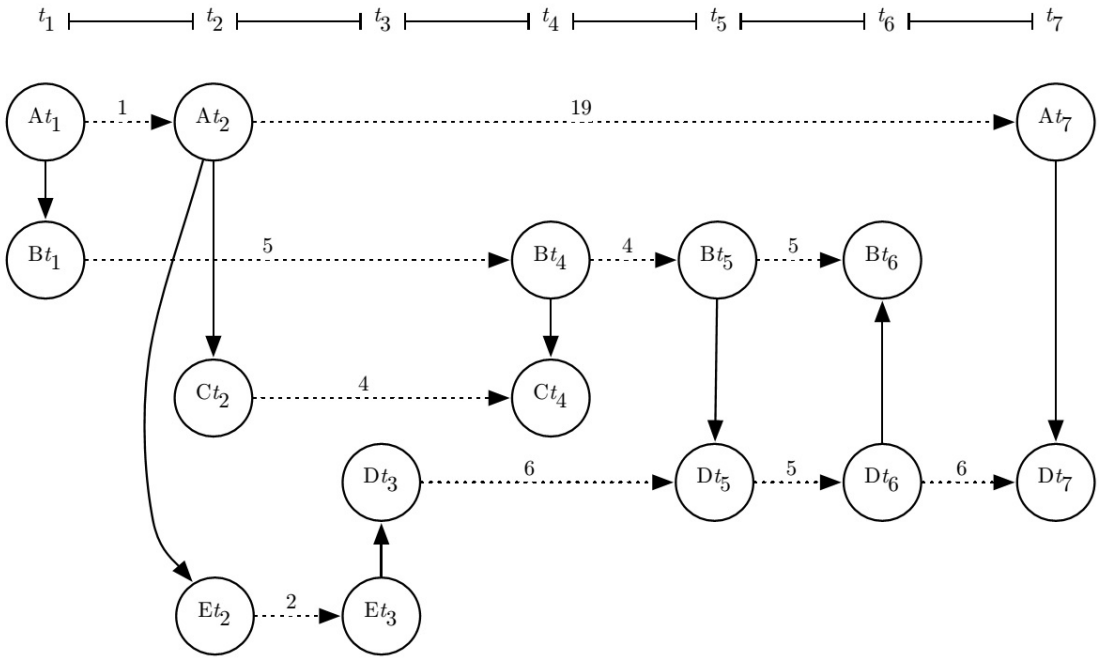


Figure 2-11: A simple temporal graph [85]

Temporal graphs have been defined as a tool for analyzing rich temporal datasets that describe events over periods of time [85]. In addition to regular edges between vertices, temporal graphs can have temporal edges to encode dynamically changing interactions. Vertices represent objects at a specific time point.

Static edges represent edges between vertices at the same time point. Temporal edges connect vertices corresponding to the same object at different time points. Figure 2-11 shows an example of a temporal graph. In the figure, the vertex A_{t_1} corresponds to the object A at time point t_1. The edge (A_{t_1}, B_{t_1}) is a static edge and (A_{t_1}, A_{t_2}) is a temporal edge. Further, the weights on the temporal edges correspond to the amount time elapsed between the two time points.

2.6.1 Graphs for HAR

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]. On the other hand, graphical models like Hidden Conditional Random Fields have been used to correlate spatial features of the video activities [51]. Researchers have also used graphical models that encode spatial and temporal features of an activity simultaneously [46, 87]. Researchers have represented the vision-based features of particular time-points during an activity using a *structured feature graph* [46]. A sequence of such graphs is then used to represent the complete activity. Human activities have also been represented as hierarchical qualitative spatio-temporal graphs [87]. Qualitative spatial relations between objects are represented using vertices of one level in the hierarchy and qualitative temporal relations as vertices at another level of the hierarchy. Graph representations of an activity based on a volumetric view of the have also been discussed where vertices represent spatio-temporal segments of the video [45]. Directed Acyclic Graphs (DAGs) have been used to describe activities wherein nodes represent motion patterns of a set of entities [88]. The nodes are linked with edges if the corresponding motion patterns are temporally related. Spatial and temporal features have also been encoded with a two-graph model [89]. One graph encodes only spatial features and the other encodes the temporal relations between the features.

However, none of the representations discussed above explicitly track the evolution of spatial relations between components of interacting objects. It is possible to achieve a similar effect using the representation discussed in [87]. However, the number of vertices will increase considerably when extended objects are used for abstraction. The DAG representation of [88] encode relations between the motion patterns of individual entities. However, the information about the motion patterns themselves are not retained. The two-graph model discussed in [89] is difficult to handle because two different graphs are used to keep spatial and temporal

information. To deal with such problems, in Chapter 4 of this thesis, a *temporal activity graph* representation of human activities is discussed.

2.6.2 Learning within Graphs

Graphs are extensively used for representational purposes in a large number of application within and beyond computer science. This calls for methods that can be used for learning from data represented as graphs. However, traditional learning algorithms can not be directly applied for graphs because they do not have have a vector representation. Learning within graphs have been researched extensively and a variety of different techniques have been developed over the years. Learning algorithms for graphs fall into two distinct categories [90]. In the first category, a single graph is seen to represent an entire network of data. That is, a complete dataset is represented by a single graph. In such a case, learning within graphs is a matter of finding repeating substructures within the same graph. In the second category, individual graphs represent individual data points within the dataset. The entire dataset is a collection of smaller graphs. For both categories, specialized methods and generic methods coupled with distance and kernel functions have been discussed in literature. *Graph based relational learning, graph grammar induction* etc are examples of specialized methods that have been adapted from classical learning algorithms. Generic methods for learning from graph usually involve designing of specialized *kernel* and *distance* functions in combination with generic supervised and unsupervised algorithms. This thesis focuses on the second category of graph learning algorithms. In the *Temporal Activity Graph* representation that is presented in Chapter 4, a single activity is represented by a single graph structure.

2.7 Graph Classification using Kernels

A generic approach for learning using graphs is using generic similarity-based supervised and unsupervised techniques together with specialized *distance* functions or *kernel* functions. Distance functions compute the dissimilarity between two graphs, i.e. a higher value indicates more dissimilarity. On the other hand, kernel functions compute similarity between graphs, i.e. a higher value indicates more similarity. Both are used by a variety of generic clustering and classification techniques. In this case, the learning problem boils down to finding appropriate distance or kernel functions for the graphs. Further, for kernel functions to be

used with generic classification techniques such as SVMs, it is essential for the kernel function to be *positive semi-definite*.

Various strategies have been discussed in literature for defining appropriate kernel function. The problem of computing similarity between two graphs can be seen as the problem of *graph-isomorphism*. Graph isomorphism is the problem of determining that two graphs have the same number of vertices connected in the same way. However, graph isomorphism is a known NP-complete problem. In order to deal with this problem, various approximate approaches to computing kernel function have been discussed. A popular category of such kernel functions are the *label-sequence kernels* [91]. In label sequence kernels, the edges and vertices of the graphs are assumed to labeled. A label-sequence is an alternating sequence of edge and vertex labels that is generated by some walk within the graph. Random walk kernels functions matches the label sequences of the two graphs, obtained by random walks, to determine their similarity. Although ideally it is preferable to compute all possible label-sequences for the two graphs, the number of label-sequences would be infinite for cyclic graphs. One simple way to overcome such a problem is by restricting the length of the label-sequences.

2.7.1 Temporal Graph Kernel

Temporal graphs are often used for modelling dynamic temporal evolution of structural properties. Consequently, several such temporal graph representations also fall back on kernel based classification. Early discussions on temporal graph analysis have focused on a small world network, i.e. a network that models temporal evolution of a small number of entities [92]. Therein, a parallel implementation of a sub-graph kernel is discussed for answering centrality and path-related queries. Temporal graphs have been used for modelling temporally evolving social networks [93]. The authors have used a temporal spectral graph kernel, that is a combination of several popular graph kernel techniques, to predict future growth of the network. This is an example of kernel used for finding repeating patterns within a temporal graph where a single graph represents an entire network. Researchers have also designed generic algorithms for tracking changes in similarity between two separate graphs, i.e. changes in static graph kernel values, when the graphs are dynamically evolving [94].

2.7.2 Graph Kernels for HAR

To the best of our knowledge temporal graphs have not been used in literature for HAR. However, static graphs are often used for modelling activities, as seen from the discussion in Section 2.6.1. Several of these graph representation for HAR are used together with a graph kernel based classification [46, 88, 89, 95]. A generalized random walk kernel for unlabeled directed graph has been presented for classification of activities represented as DAGs [88]. Context Dependent Graph Kernels have been presented for static attribute graphs that compute similarity based on primary walk groups (PWGs) [89]. The two graph are decomposed into PWGs and their kernel is computed by a context-dependent matching of their PWGs. A random-walk based kernel that combines subgraph matching and time sub-sequence matching for computing similarity between temporal sequences of graphs is discussed in [46]. A path-based graph kernel has also been discussed in the context of human behavior analysis [95].

Graph kernels discussed within HAR are designed for static graphs, whereas activities are represented using temporal graphs in this work. Further, the temporal graph kernels discussed in Section 2.7.1 are insufficient for handling the spatio-temporal structure of TAGs. Therefore, a *Temporal Activity Graph Kernel* (TAG kernel) is presented in Chapter 4 of this thesis. The TAG kernel computes similarity between two TAGs and is based on the similarity between *label sequences*. However, the label sequences discussed herein differs from what has been discussed in literature. Here, label sequences are determined by the temporal evolution of spatial relations between components of interacting objects. Furthermore, the label sequences are sequences of qualitative spatial relations between interacting entities and their similarity depends on qualitative relational properties.

2.8 Grammar based Recognition

Four types of grammar are recognized in formal language theory based on their expressiveness [96]-

(a) Type 3 grammars for *regular languages*,

(b) Type 2 grammars for *context free languages*,

(c) Type 1 grammars for *context sensitive languages*, and

(d) Type 0 grammars for *recursively enumerable languages*.

Type 0 and Type 1 grammar are more expressive and cover a wider range of languages. However, known parsing algorithms for Type 0 and Type 1 grammars have exponential time dependency [97]. On the other hand, Type 2 grammars or *context free grammars* have polynomial-time parsing algorithms. Furthermore, context free grammars are less restrictive than Type 3 or regular grammars. For this reason, context free grammars are often used for representational purposes. In literature, there exist reports of researchers who prefer grammar as a tool for encoding the recursive and hierarchical nature of human activities [6, 52, 53, 54]. Some of these works represent the hierarchical nature of human activities using a Context Free Grammar(CFG) [6]. The researchers have used lower level gestures as terminals for the CFG, and activities are seen as a string of gestures with time-based constraints. They further present a heuristic-based parsing but rely on grammar rules that are hand-coded by a human expert.

Stochastic Context Free Grammars (SCFG) extended with temporal relations have been used to encode activities as parallel strings of gestures [52]. For example, raising both hands simultaneously, could be interpreted as an action consisting of the parallel *sub-events* of raising the left hand and raising the right hand. The grammar rules are learned automatically from training data and a multi-thread parsing algorithm is proposed to handle concurrently occurring strings of gestures.

AND-OR Grammars have also been discussed in the context of HAR [53, 54]. Researchers have defined AND rules to describe sequences of action primitives for an activity and OR rules to describe alternate sequences for an activity [53]. Although the AND-OR grammar rules are initially learned using a manually labeled semantic map of the scenes, a method has been proposed to discard the manual labels. Stochastic AND-OR grammars have also been learned in an unsupervised setting and used for activity recognition [54].

It is notable that although grammars have often been used for HAR, the structure of the grammar depends largely on the representation of activities used in the respective work. In this work we present a *Temporal Activity Graph* representation of activities in Chapter 4. Based in this representation, in Chapter 5 a *Temporal Activity Graph Grammar* is proposed for modelling human activities. A graph-grammar induction algorithm is presented for learning the grammar rules from a set of positive examples.

2.8.1 Context Free Grammars and Parsing

A context free grammar (CFG) [98] is defined as a four-tuple $\langle \Delta, \Sigma, R, S \rangle$ where,

- Δ is the set of variables or non-terminals
- Σ is the set of terminals
- R is the set of rules, such that each rule is of the form $A \rightarrow \alpha$, where $\alpha \in (\Sigma \cup \Delta)^*$
- S is the start symbol, and $S \in \Delta$

The set of strings that generated by the rules of grammar comprise the *language* of the grammar. Conversely, a grammar can be seen as a condensed description of a set of strings i.e. a language [97]. The problem of deciding whether any string of terminals, $w \in \Sigma^*$ belongs to the language of a grammar or not is called the *recognition problem*. On the other hand, *parsing* not only gives a yes or no answer to the recognition problem, it further reconstructs the sequence of steps to produce the string from the given grammar if it is a member.

For parsing strings given a Context Free Grammar, several parsing algorithms have been developed over the years. LL(k) parser [99], LR(k) parser [100], CYK parser [101], Earley parser [102] are some of the notable parsing algorithms for Context Free Grammars. LL(k) parsers are top-down parsers that scans the input string from left-to-right, uses only left-most derivations, and uses k lookahead tokens in the string [99]. However, LL(k) parsers are not able to parse all context-free languages.

LR(k) parsers are bottom-up parsers that scan the string from left-to-right, uses only right-most derivations, and uses k lookahead tokens [100]. LR parsers scans the string and deterministically applies two types of basic operations called *shift* and *reduce*. The *shift* operation simply instructs the parser to move on to the next symbol on the input string. The *reduce* operation instructs the parser to apply some grammar rule. A rule of the form $A \rightarrow \alpha$ is applied when the rightmost subsequence in the string is same as the string α. Such a reduce operation essentially replaces the subsequence α with A. Implementing LR(k) parser involves constructing a LR parse table. The entries of the table determine whether a shift or a reduce operation is to be applied at any given point of time during the parsing. The entire parsing process involves a table look-up and accordingly performing a shift or a reduce operation. There exist automated methods for constructing the PR tables called *parser-generators*. Compared to LL(k) parsers,

LR(k) parsers can be used for a larger number of context-free languages. However, LR(k) parser can be used only for a subset of context-free languages called *deterministic context-free languages*.

The CYK and Earley parsing algorithms are more generic than LL(k) and LR(k) parsing algorithms because they can be used for *any* context free language. CYK parsing can be applied only for grammars that are in *Chomsky Normal Form* but all context free grammar can be easily converted to CNF. On the other hand Earley parsing algorithm is known to be the most efficient parsing algorithm that also does not require grammars to be in any particular form.

2.8.2 Probabilistic Context Free Grammar

A Probabilistic Context Free Grammar (PCFG) or Stochastic Context Free Grammar is a context free-grammar that has probabilities associated with each rule in R. It was first defined for recognizing RNA sequences in bioinformatics [103]. The probabilities are assigned to the rules such that the sum of probabilities of all rules, with some $V_i \in \Delta$ on the left hand side, is 1. A PCFG consists of the same components as context free grammar but contains an additional probability function.

A PCFG is quintuple $\langle \Delta, \Sigma, R, S, Pr \rangle$ where, Δ, Σ, R and S are the set of non-terminals, set of terminals, set of rules and the start non-terminal respectively. Pr is the probability function that assigns a probability value to each rule $A \rightarrow \alpha \in R$ such that,

$$Pr(A \rightarrow \alpha \geq 0) \quad \text{and} \quad \sum_{\alpha} Pr(A \rightarrow \alpha) = 1$$

$Pr(A \rightarrow \alpha)$ is interpreted as the probability of choosing rule $A \rightarrow \alpha$ to replace the variable A in a derivation.

2.8.3 Graph Grammar

Similar to regular string grammars, a graph grammar defines a set of production rules that allow one to construct a specific set of graphs. A production rule in a graph grammar is of the form $X \rightarrow D$ where X is a non-terminal and D is an undirected graph with terminals or non-terminals. D is often called the *daughter* graph. The graph in which X appears so that the rule $X \rightarrow D$ can be applied is called the *mother* graph. Depending on whether the terminals represent a *node* or an *edge* in the mother graph, a graph grammar can be either a *node-replacement grammar* or an *hyperedge-replacement grammar*.

A node replacement grammar with a *neighbourhood controlled embedding* is a system $G = (\Sigma, \Delta, P, S)$. Here Σ and Δ are the set of terminal and non-terminal node labels respectively. P is the set of productions and S is the start non-terminal. Every production of the form $X \rightarrow D$ is associated with a connection relation that defines which node of D is connected to which nodes in the mother graph. A simple connection relation can be seen as a pair (x, y) where x is a node in the daughter graph D and y is a node in the mother graph. The semantics of such a connection rule is that an undirected edge exists between x and y when X is replaced by D in the mother node. The process of replacing X by D and making new edge connections is called *embedding*.

In hyperedge replacement grammar, we deal with hypergraphs that consist of sets of nodes and sets of hyperedges. Hyperedges of type-k are a structure with k tentacles that are able to connect k nodes (see Figure 2-10e). The terminals and non-terminals represent hyperedges. Non-terminals representing a hyperedge can be attached to any structure with a set of nodes by attaching each of its tentacles to a node. Here too, every rule is of the form $X \rightarrow D$ where D is a subgraph with hyperedges that has a designated set of *external* nodes. When replacing X during a derivation the *external* nodes of D are *glued* to nodes that were connected by the hyperedge X in the mother graph.

2.8.4 Graph Grammar Induction

Grammar induction is the process of constructing a set of grammar rules that generate a given set of positive examples. Grammar induction has been extensively studied for string grammars and a large number of grammar induction algorithm have been proposed [104, 105]. Relatively fewer work has been done in the context of graph grammars. Nonetheless, there are several reports of work on graph grammar induction and its applications. One of the earliest examples for graph grammar induction algorithms induces a special category of context-sensitive graph grammars [106]. Since then several generic and application specific graph grammar induction algorithms have been proposed. SubdueGL is a popular generic graph grammar induction algorithm that discovers frequent substructures based on the concept of minimum description length[107]. SubdueGL learns a context free graph grammar in which variables appear as node/vertex labels. A minimum description length based algorithm has also been used for learning Stochastic Context Free Graph Grammars [108]. Generic algorithms have also been presented for inducing non-confluent graph grammars, i.e. grammars where structure of the re-

sultant graph depends on the order in which the rules are applied [109]. Grammar induction algorithms that look for frequent subgraphs using constraint programming local consistency techniques have also been presented [110]. An alternative Plane Graph Grammar was introduced in [111]. The researchers further discuss an appropriate graph grammar induction algorithm for learning rules of a Plane Graph Grammar.

On the other hand, specialized graph grammars have often been designed for modelling in various applications. A graph grammar induction algorithm based on overlapping subgraphs has been presented that was intended for modelling Visual Programming Languages [112]. Recent reports have also often discussed genetic programming techniques for learning graph grammar rules [113, 114]. Genetic programming has been used for inducing the rules of a Context Sensitive Graph Grammar [113]. Evolutionary Computing techniques are also used for inducing rules of an Augmented Graph Grammar [114]. Augmented Graph Grammars are analogous to string grammars but also allows for graph structure and have been discussed in the context of argument analysis [115]. Context Free Geometric Graph Grammars have been introduced for modelling and synthesis of urban road networks [116]. The work further presents a grammar induction algorithm that determines frequently repeating subgraphs by detecting isogroups.

It is clear from the discussion above that, while there exist generic graph grammar induction algorithms [107, 108, 109, 111], these may not be applicable for many specialized graph grammars introduced for the specific application areas. Most applications that use graph grammars incorporate features specifically designed to serve a purpose [112, 115, 116]. The generic algorithms are unable to take advantage or avoid the disadvantages of such features. Therefore, induction algorithms that are specially designed for a particular category of graph grammars are often used. In Chapter 5 of this thesis, a *Temporal Activity Graph Grammar* is presented for modelling human activities along with an induction algorithm. The induction algorithm takes advantage of the unique structure of Temporal Activity Graphs. The Temporal Activity Graph Grammar presented herein is a probabilistic context free grammar with elements of node-replacement graph grammar.

2.9 Conclusion

In this chapter a discussion on the domain of Human Activity Recognition and related work that has been done in the field has been presented. Further, basic background knowledge on which the remaining chapters of this thesis are built are

also discussed. In the next chapter, the first component of the thesis is presented which is a geometric framework for extracting qualitative relations for a pair of extended objects.