# Chapter 2

# Literature Review

## 2.1 Introduction

In this chapter, we would like to present brief discussions on important concepts of qualitative spatial reasoning and grammar based pattern recognition. Moreover, approaches for motion pattern recognition in computer vision as well as in Gi Science are outlined.

## 2.2 Qualitative Spatial Reasoning

### 2.2.1 What is Qualitative Spatial Reasoning (QSR)?

Qualitative spatial reasoning is a knowledge representation and reasoning technique. It is used for representation of commonsense knowledge without resorting to numerical values [8]. Quantitative information is precise and accurate, but in many applications, precise quantitative representation may not be necessary. Human cognition is often qualitative. In systems where we need to represent knowledge at a level close to human cognition, qualitative representation may be a better idea.

Early attempts at developing qualitative reasoners for spatial and kinematic mechanisms were not successful. This led to what is known as the poverty conjecture [17]. It was argued that there can be no purely qualitative spatial reasoning mechanism. Forbus et. al. felt that the best way to overcome these limitations is to combine quantitative and qualitative representations [17]. Space is multidimensional and there was a doubt whether transitivity of qualitative spaces is possible in higher dimensions. It was concluded that representations in higher dimensions are sparse and spatial reasoning in higher dimensions can not be done without resorting to numbers. Over the years, many qualitative constraint calculi have been proposed for different aspects of space. The most important aspects are topology, orientation and distance. Other aspects are size, shape, morphology and spatial change. Most works on qualitative spatial reasoning have focused on single aspect of space.

Space and time are two important aspects of our commonsense knowledge. Time is a scalar quantity. Qualitative temporal reasoning has emerged as a sub field of qualitative reasoning. Allen's interval algebra (IA) [18] is an important work in qualitative temporal reasoning. IA defines thirteen Jointly Exhaustive and Pairwise Disjoint (JEPD) binary qualitative relations that may hold between a pair of convex time intervals. Representation of space is more complex because of its multidimensionality. Multidimensionality of space becomes apparent from the qualitative terms that we use in natural language to refer to spatial aspects. For example, in everyday description of spatial events, we use terms like *inside*, *outside* etc. for topology, terms like *left*, *right* for direction, *close* , *far* etc. for distance.

In QSR, a set of basic binary qualitative relations is used for representation of knowledge. This set partitions the domain under consideration. Moreover, between any two objects, only one of the relations can hold. In other words, the set of binary qualitative relations is Jointly Exhaustive and Pairwise Disjoint (JEPD). A partition of a set is a grouping of the sets elements into non-empty subsets, in such a way that every element is included in one and only one of the subsets. A set of Jointly Exhaustive and Pairwise Disjoint (JEPD) relations is same as a partition. In QSR literature the term JEPD is more prevalent. Similarly, the term *complete* conveys the same meaning as *Exhaustive*. For example, let us consider the distance between two objects. Distance is continuous and can assume any positive real value. We discretise this continuous domain into qualitative categories like *close*, *near* and *far*. Any possible distance between the objects will belong to one of this qualitative categories. Therefore, the set of qualitative relations { *close*, *near*, *far* } is Jointly Exhaustive. This set is Pairwise Disjoint also because any numerical distance between the objects belongs to exactly one of the relations. Similarly, a *JEPD* set for comparing relative size of two objects may be { *equal*, *bigger*, *smaller* }.

In QSR there are two forms of reasoning. One is based on construction of composition tables [19]. Composition tables are usually pre-computed from the semantics of the basic relations. If the binary qualitative relation *r1* holds between objects $A$ and $B$ and the relation *r2* holds between objects $B$ and $C$, then the relation between $A$ and $C$ can be found by computing the composition of *r1* and *r2*. This result is stored in a composition table in the row indexed by *r1* and in the column indexed by *r2*. Composition table for RCC-8 [20] is presented in the work of Wolfl et al. [21].

The other form of reasoning is based on the notion of spatial change. An inherent assumption in QSR is that change is continuous [22]. In order to change from one qualitative value to another, all the values in between must be passed through. The set $\{\ close, near, far\ \}$ of binary qualitative relations for distance can be taken as an example. The distance between two objects can not directly change from *close* to *far*. It has to change from *close* to *near* and then finally to *far*. This notion of spatio-temporal continuity in expressed using a graph known as conceptual neighbourhood graph. A relation *r2* is a conceptual neighbour of another relation *r1* if we can arrive at *r2* from *r1* by continuous change [23]. In the set of distance relations, *near* is a conceptual neighbour of *close* and similarly *close* is a conceptual neighbour of *near*. If two objects are close and if the distance between them increases, then the binary qualitative distance relation will change from *close* to *near*. On the other hand, if the distance decreases and the current relation is *near*, then we arrive at the relation *close* next. In a conceptual neighbourhood graph, vertices represent binary qualitative relations. An edge is drawn from one vertex to another if the qualitative relation represented by the second vertex is a conceptual neighbour of the qualitative relation represented by the first vertex. Using this notion of spatio-temporal continuity, we can reason about changes that are possible.

### 2.2.2 QSR Formalisms for Orientation, Direction and Distance

#### Direction and Orientation

Direction is an important concept in qualitative spatial formalisms. Qualitative direction relations describe the qualitative direction of one object with respect to another. A frame of reference is necessary for expressing qualitative directions [24]. Therefore, qualitative direction is based on three concepts, namely, primary object, reference object and a frame of reference (FoR). Some qualitative direction relations reported in the literature use triadic relations, taking into account these three aspects. Other formalisms use external reference directions. For example, the geographical direction *east*, *west*, *north* and *south* can be used as a frame of reference. Such an external reference frame is known as extrinsic or allocentric frame of reference. In intrinsic or egocentric frame of reference, an object is directed along an intrinsic direction that depends on properties like its shape, size etc. This intrinsic direction sets up a reference system and in this case, direction labels like *front*, *left*, *back*, *right* etc. are typically used to convey the directional information. In a deictic frame of reference, direction of a primary object with respect to a reference is expressed from the point of view of an external observer [25]. Qualitative direction and orientation are closely related. Spatial orientation expresses the spatial location of a primary object with respect to a reference object. When we say that an object $A$ is to the north of another object $B$, we convey information about cardinal direction as well as spatial orientation. Similarly, a qualitative term like *front* has a flavour of direction and orientation.

Spatial objects are often abstracted as dimension less points, one dimensional

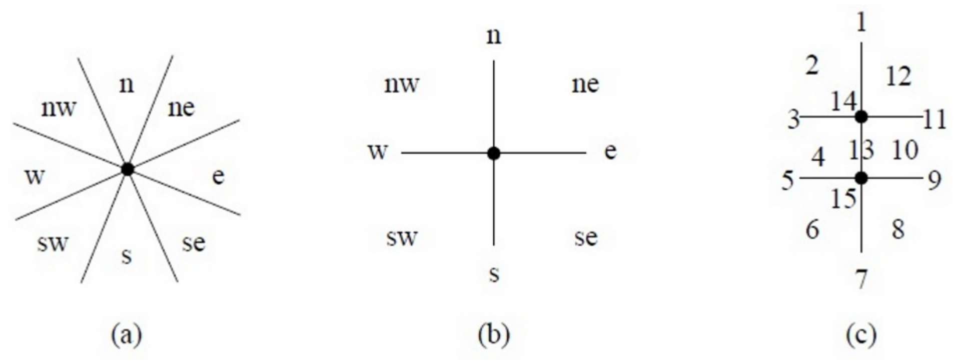


Figure 2.1: Cone Based and Projection Based Orientation

directed line segments and also as extended spatial entities. Most of the works on direction and orientation, reported in literature, abstract objects as points. Frank proposed different methods for expressing the cardinal direction of a primary point with respect to a reference point in geographical space [26]. Labels like *north*, *south*, *east* and *west* etc. express the orientation of a primary object with respect to the reference. Frank suggested a cone based and a projection based approach. In part (a) and (b) of Figure 2.1, these cone based and projection based orientation models are shown. Orientation relations are binary because an allocentric FoR is used in these models.

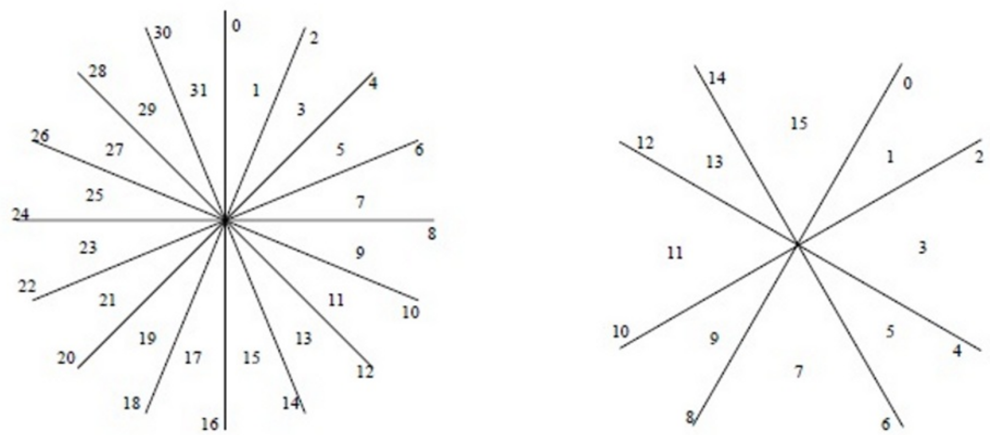

Figure 2.2: Star Calculus: Renz and Mitra

In the work of Frank, eight orientation labels,namely, *n*, *s*, *e*, *w*, *ne*, *nw*, *se* and *sw* are used for expressing the orientation of a primary point with respect to the reference point located at the centre. These calculi were generalized into a

*Star Calculus* by Renz and Mitra [27]. This calculus is based on $n$ number of lines $l_i$ with given angles $\delta_i$ that define *2n* sectors and *4n+1* basic relations (shown in Figure 2.2). Two different star calculi, one with 8 lines and 33 relations and the other with 4 lines and 17 relations are shown in the figure. The granularity can be adjusted by changing the number of lines and the angles of the sectors. *Star Calculus* uses an allocentric frame of reference.

Freksa proposed another projection based model for point objects which is known as double-cross calculus [23]. This calculus defines the direction of a located point to a reference point with respect to a perspective point. Here, three axes are used. One axis is obtained by connecting the perspective point to the reference point and the other two axes are perpendicular to this line and intersects it at the positions of the perspective point and the reference point. This is shown in part (c) of Figure 2.1. In double-cross calculus, fifteen ternary basic orientation relations are defined.

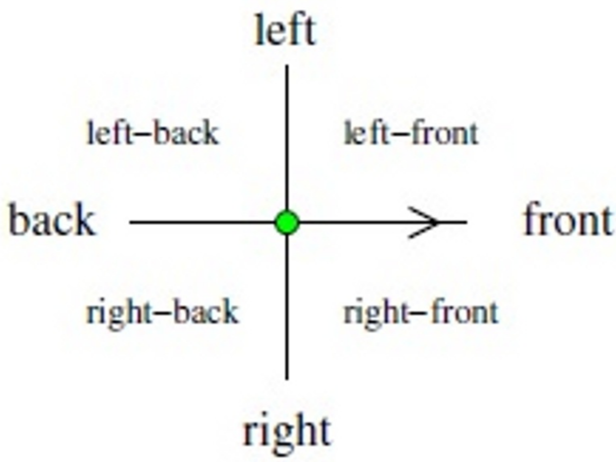


Figure 2.3: Direction of An Oriented Point

There are other formalisms for direction and orientation that abstract spatial objects as directed entities. Notable works include oriented point relation algebra (*OPRA*) by Moratz [28] and dipole relation algebra by Moratz, Renz and Wolter [29]. In *OPRA*, basic entities are points that have an intrinsic direction. In Figure 2.3, intrinsic direction of a point is shown.

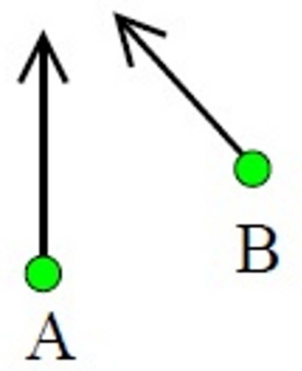


Figure 2.4: Orientation Relation Between Two Points

This direction of a point sets up a reference system and divides the plane into different direction sectors. For defining basic orientation relations, two labels are used. The lower label tells the qualitative spatial orientation of the second point when seen from the first point. The upper label expresses the orientation first point as seen from the second. An orientation relation between two oriented points is shown in Figure 2.4. The lower label for this relation is *rightfront* (abbreviated as *rf*) and the upper label is *leftfront* abbreviated as *lf*. Therefore, if the name of the first object is $A$ and that of the second object is $B$, then the relation is $A \ \frac{lf}{rf} \ B$.

In dipole relation algebra, spatial objects are abstracted as directed line segments (named as a dipole). A dipole has two end points. For any dipole $A$, its start point is denoted as $S_A$ and its end point is denoted as $E_A$.

In Figure 2.5, two dipoles $A$ and $B$ and the *lrrr* orientation relation that holds between these two dipoles is shown. The first two letters of this relation i.e. $l$ and $r$ express the relative position of the end points of the dipole $B$ with respect to $A$. The point $S_B$ lies to the left and the point $E_B$ lies to the right. The last two letters represent the relative positions of the end points of $A$ with respect to $B$. The point $S_A$ as well as $E_A$ lie to the right of the dipole $B$. Therefore, the orientation relation is *lrrr*.

Dipole calculus can be defined at different granularities. In Figure 2.6, atomic orientation relations of a coarse grained dipole calculus are presented.

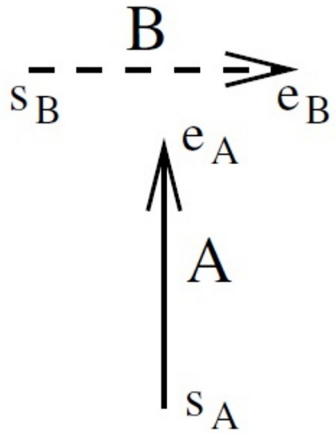


Figure 2.5: lrrr Relation Between Two Dipoles

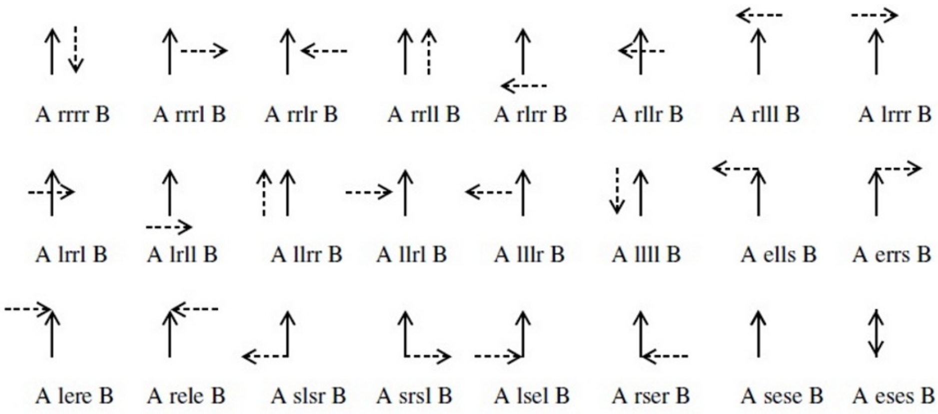


Figure 2.6: Twenty Four Atomic Relations of Coarse Dipole Calculus

It is much more difficult to define orientation relations for spatial entities that are extended in space. Problem is more complicated when these objects have holes or if they are multipiece. In QSR literature, we find that many researchers abstract all such extended regions as rectangles whose sides are parallel to the axes of projection in a two dimensional plane. When this type of abstraction is used, it is possible to represent an extended region by its projections to the axes (shown in part(a) of Figure 2.7). The orientation relations between two such rectangles can be expressed using Allen's interval algebra (*IA*) relations. For each such rectangle, there are two projections. Then, *IA* relations can be defined between corresponding projections. Balbiani et. al. defined *13* × 13 basic relations for two rectangles and the resulting calculus is known as rectangle

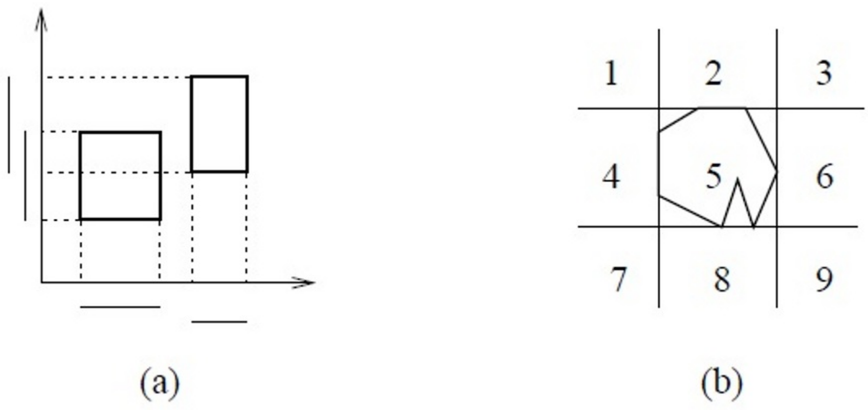


Figure 2.7: Rectangle Algebra and Direction Relation Matrix

algebra [30].

Goyal and Egenhofer [31] proposed a model for representation of orientation of rectangular objects. The sides of rectangles are parallel to the axes of projection. An object is represented by its Minimum Bounding Rectangle (*MBR*) and the sides of this reference *MBR* are extended as shown in part (b) of Figure 2.7. This results in nine sectors. The primary rectangle may be contained in one of these sectors or it may span more than one sector.

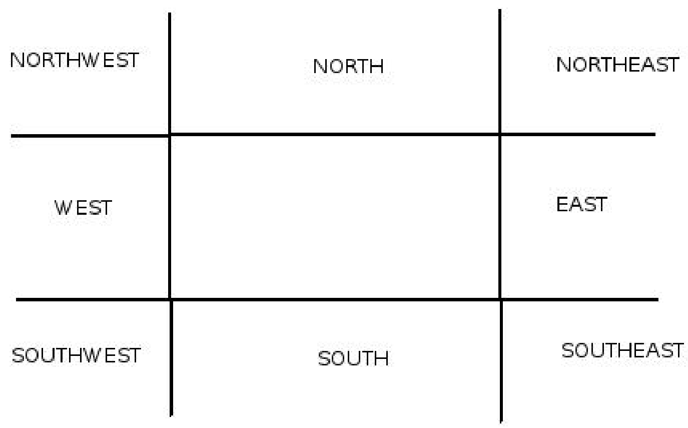


Figure 2.8: Rectangular Cardinal Directions

Skiadopolous and Koubarakis developed reasoning algorithms for this calculus and studied its computational properties [32]. This work was taken further by Sciavicco and Navarrete [2]. Instead of using numbers for sectors, they assigned labels like *North*, *South*, *NorthWest*, *SouthWest* etc. and proposed a set of rectangular cardinal directions for rectangular spatial objects. Each rectangular cardinal direction relation can be mapped to a set of rectangle relations proposed in Balbiani's work. In Figure 2.8, the unlabelled middle rectangle is the *MBR* of the reference object. The primary rectangle may have relations like *NorthWest*, *NorthWest:North:NorthEast*, *East:SouthEast* etc. with respect to the reference rectangle.

## Distance

Distance is an important aspect of space. In our everyday communication, we often refer to distance in some way or other. Sometimes we use qualitative terms like "Place A is close to place B" to convey our idea of distance. Use of quantitative information like "I am 1 k.m. away from my work place" is also common. Distance expressions can be relative too. Sentences like "I am nearer to place A than to place B" are often heard. Qualitative distance relations can be absolute or relative [8]. In absolute distance relations, the exact distance between the objects will be considered. Otherwise, distance relations may be relative. In QSR literature, most works on qualitative distance abstract spatial objects as points. Absolute distance relations can be defined at various granularity levels [33]. This issue of granularity depends on the scale of space. In defining these relations, the real number line is divided into various zones and qualitative distance labels are assigned to each zone. At a coarse granularity, these labels may be *close*, *near* and *far*. At a finer level, these may be *very close*, *close*, *commensurate*, *far*, and *very far*. For relative distance representation, ternary relations like *closer than*, *farther than* etc. are often used. Distance and orientation are intricately related and qualitative distance is often combined with orientation information. Clementini et. al. have termed this kind of information as *positional information*. They have combined a cone-based orientation approach with absolute distance relations and presented different procedures for computing the composition of two positional relations (A,B) and (B,C)ClementiniFeliceHernandez. Isli and Moratz have proposed several position calculi on various levels of granularity by combining relative distance relations with different approaches to orientation such as the projection-based approach or the double-cross calculus [19].

Size is another aspect that can be represented qualitatively. An early work in this area that combines translation with rotation is reported in [34]. Using this model, it is possible to compare relative sizes of size and shape invariant regions. Here, if translation is possible and after translation, a region becomes proper part of another, then it is argued that the size of this object must be smaller.

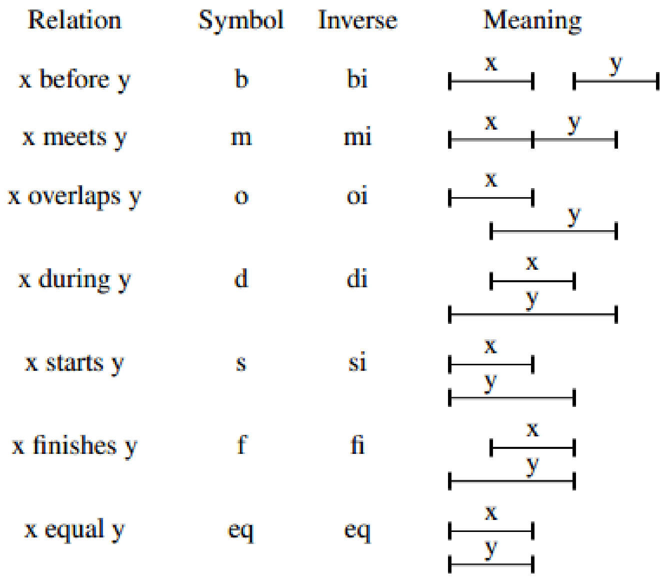


Figure 2.9: Allen's Interval Algebra Relations

### 2.2.3 Allen's Interval Algebra

On the temporal dimension, a notable work is that of James Allen [18]. Allen proposes to represent temporal relations by defining thirteen mutually exclusive qualitative relations between intervals of time. These relations are shown in Figure 2.9. The set consists of seven basic relations and their inverses. For the *equal* relation, the converse is same as the basic relation. In Figure 2.9, each interval of time is shown by a line segment. Relation names have been abbreviated. Temporal reasoning is done by deriving compositions of basic relations and storing them in a table. Composition may result in unique relations or a set of relations. For example, let $A$, $B$ and $C$ be three intervals of time. We assume that the relation between $A$ and $B$ is *meets* and that between $C$ and $B$ is *during*. The relation between $B$ and $C$ is *during inverse* ( *di* ). It can be inferred that the interval $A$ must come before the interval $C$. So, the composition of $A$ and $C$ gives the relation *before*.

### 2.2.4 Spatio-temporal Continuity

In spatio-temporal representation and reasoning, change is an important notion [35]. Change can be continuous or discontinuous. In discontinuous changes, in any sufficiently small temporal neighbourhood, an attribute having ordered values will change from one value to another without taking on all intermediate values [36]. When change is continuous, intermediate values are also passed through. As an example, we can consider distance between two objects. On quantitative scale, this distance is a real number interpreted in some unit. If the objects are far, the numerical distance value is high and when they are close, this value is low. Distance can not suddenly jump from this high value to the low value. When objects move closer, distance value will gradually diminish and finally reach the low value. This idea of spatio-temporal continuity can be an important concept in qualitative reasoning [8]. Forbus [22] has emphasised the importance of spatio-temporal continuity as *A simple consequence of continuity, respected by all systems of qualitative physics, is that, in changing, a quantity must pass through all intermediate values. That is, if $A < B$ at time t1, then it cannot be the case that at some later time t2 $A > B$ holds, unless there was some time t3 between t1 and t2 such that $A = B$.*

Conceptual neighbourhood of relations [37] can be considered as a method of representing continuous change. Freksa has defined conceptual neighbourhood as "Two relations between pairs of events are conceptual neighbors if they can be directly transformed into one another by continuous deformation (i.e., shortening or lengthening) of the events". Continuous deformation may involve different activities like rotation, translation, scaling, movement etc. For example, in Allen's interval algebra, the relations *before* and *meets* are conceptual neighbours because when the *before* relation is extended in time, it results in the *meets* relation. Conceptual neighbourhood graphs are used to express conceptual neighbourhood of relations. Nodes in a conceptual neighbourhood graph repre-

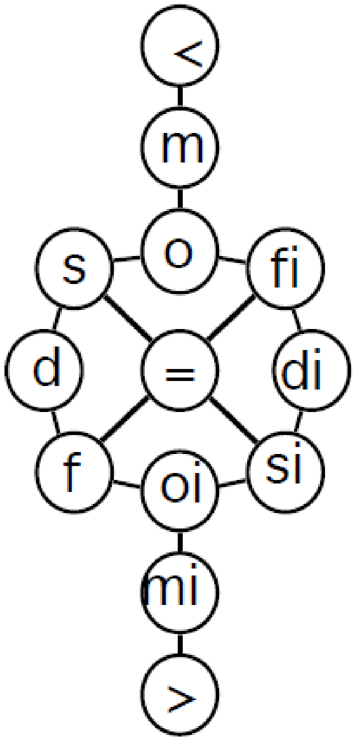


Figure 2.10: Conceptual Neighbourhood Graph: Allen's Interval Algebra

sent spatial or temporal relations. Edges are created to connect the relations that are conceptual neighbours. Conceptual neighbourhood graphs express the fact that some relations are closer to each other than others. Such a graph for Allen's interval algebra is shown in Figure 2.10. Conceptual neighbourhood graphs can be used to find alternate possible paths between two relation nodes. These paths convey information about sequences of deformations because of which the change in relation occurred. When we are given any two relations in the graph, the neighbourhood graph helps in knowing the possible change process that brought about the change.

## 2.3 Motion Patterns Analysis in GIScience

### 2.3.1 Outline of Approaches

In GIScience, analysis of motion patterns has been an active area. GPS, mobile phones and radio transmitters have helped collection of large amount of movement data. Analysis of such mobility data helps in extraction of meaningful patterns that convey high level information to a human analyst. An early work on movement pattern analysis includes the simulation study of human adaptive behavior [38]. Imfeld performed a spatio-temporal analysis of movement data of animals [39]. There has been ample research on discovery of similar trajectories or clusters. Data for such analysis are typically location of moving point objects over consecutive time points. Laube and Imfeld proposed the REMO (RElative MOtion) framework to define motion patterns of groups of objects [40]. They used direction of motion or change of direction as the principle feature for representing spatio-temporal patterns. Laube et. al. augmented the framework by including locational information also. Spatio-temporal patterns like *flock*, *leadership*, *convergence*, and *encounter* were defined at a given time step or interval. Recognition of these patterns was algorithmic.

Noyon et. al. used relative position and relative velocity for representing motion event of a primary object with respect to a reference [41]. Gundmundsson et. al. [42] evaluated the computational efficiency for detecting four spatio-temporal patterns, namely, *flock*, *leadership*,*convergence* and *encounter* defined in the work of Laube and others. Relative orientation of points is defined in work of Mossakoski and Moratz [43]. In the work of Nico van de Weghe et. al. [9], changing distance between a pair of moving points is used for representation of motion. In their work, motion was represented within the framework of a qualitative spatial algebra. Gottfried considered relative directions and relative positions between two oriented line segments [44]. The 9+ intersection model, proposed by Kurata and Egenhofer, consider directed line segment in relation to regions for describing motion [45].

Dodge et. al. proposed a taxonomy of motion patterns for individual or groups of moving point objects [1]. Using such a taxonomy, we can define various types of motion patterns. Galton [46] augmented this work by associating specific collectives with typical motion patterns of such collectives.

In a work of Nico van de Weghe, a qualitative formalism has been used for motion representation [36]. Qualitative Trajectory Calculus (QTC) [9] makes comparisons between positions of moving objects at different time points and considers changing distance between the objects for defining the qualitative relations. Attributes of trajectories are represented qualitatively in the work of Tales et. al. [47]. In [48], integration of cross-scale analysis in spatial and temporal domains in proposed for classification of behavioural movement. Gottfried [10] defined a set of sixteen atomic motion patterns that form a relation algebra. In his work, Gottfried suggested the use of formal grammars for recognition of motion patterns.

Since we have shown the application of our language based framework for representation of the taxonomy proposed by Dodge et. al. [1], in the next section we would like to give an introduction to this taxonomy.

### 2.3.2 A Taxonomy of Motion Patterns

In the geographic domain, movement is defined as change in position of an object with condition that the identity of the object is maintained. Movement data typically are in the form of positions of an object at different time points. Movement parameters specify the features that we use to describe movement patterns. There are three types of parameters. These are *primitive parameters*, *primary derived parameters* and *secondary derived parameters* [1]. Parameters are organized along spatial, temporal and spatio-temporal dimensions. Primitive spatial parameter is the *position* of an entity. Primitive temporal parameters can be an instance of time or an interval of time. Among the primary derivatives, distance and direction of movement are in the spatial dimension and solely a direct function of position. Duration is defined as a period of time in which a movement is observed. Duration is a direct function of time and consists of one or more time intervals. Speed (i.e. rate of change of the objects position) and velocity (i.e. rate of change of position and direction) are parameters that combine both space and time dimensions, and can be derived directly from spatial position and time instances [1]. Higher order parameters of movement such as acceleration can be derived from primary derivatives. Of the secondary derivatives, the definition of the spatial parameters is assumed to be commonly known. For instance, spatial distribution represents a snapshot of the positions of *MPO*s in the database at a specific time. Sinusoity is a function of distance and refers to the degree of windingness of an objects trajectory. Among the temporal parameters, temporal distribution denotes the distribution of events along the time line. Change of duration denotes the rate of change of the duration between different observations of the same movement behavior (e.g. rate of change of the migration duration of a species of animal).

| Parameters / Dimension | Primitive | Primary derivatives | Secondary derivatives |
|---|---|---|---|
| Spatial | Position *(x,y)* | Distance *f(posn)* | Spatial distribution *f(distance)* |
| | | Direction *f(posn)* | Change of direction *f(direction)* |
| | | Spatial extent *f(posn)* | Sinuosity *f(distance)* |
| Temporal | Instance *(t)* | Duration *f(t)* | Temporal distribution |
| | Interval *(t)* | Travel time *f(t)* | Change of duration *f(duration)* |
| Spatio-temporal (x, y,t) | — | Speed *f(x,y,t)* | Acceleration *f(speed)* |
| | | Velocity *f(x,y,t)* | Approaching rate |

Figure 2.11: Movement Parameters (as defined in [1])

Acceleration (i.e. rate of change of the objects speed) represents a spatio-temporal parameter derived from speed. Approaching rate is a function of speed and distance and describes whether and how intensively a moving object approaches its destination. These parameters are shown in the Table 2.11. Generic movement patterns are common to different groups of Moving Point Objects (*MPO*) whereas behaviour patterns are specific to a certain type of *MPO*s [1].

Movement patterns are divided into generic patterns and behavioural patterns. Generic patterns are building blocks from which other patterns can be constructed. Genericity implies that these patterns are not specific to a particular kind of moving point objects; rather these patterns express commonality in movement patterns among multiple types of moving point objects. A behavioural

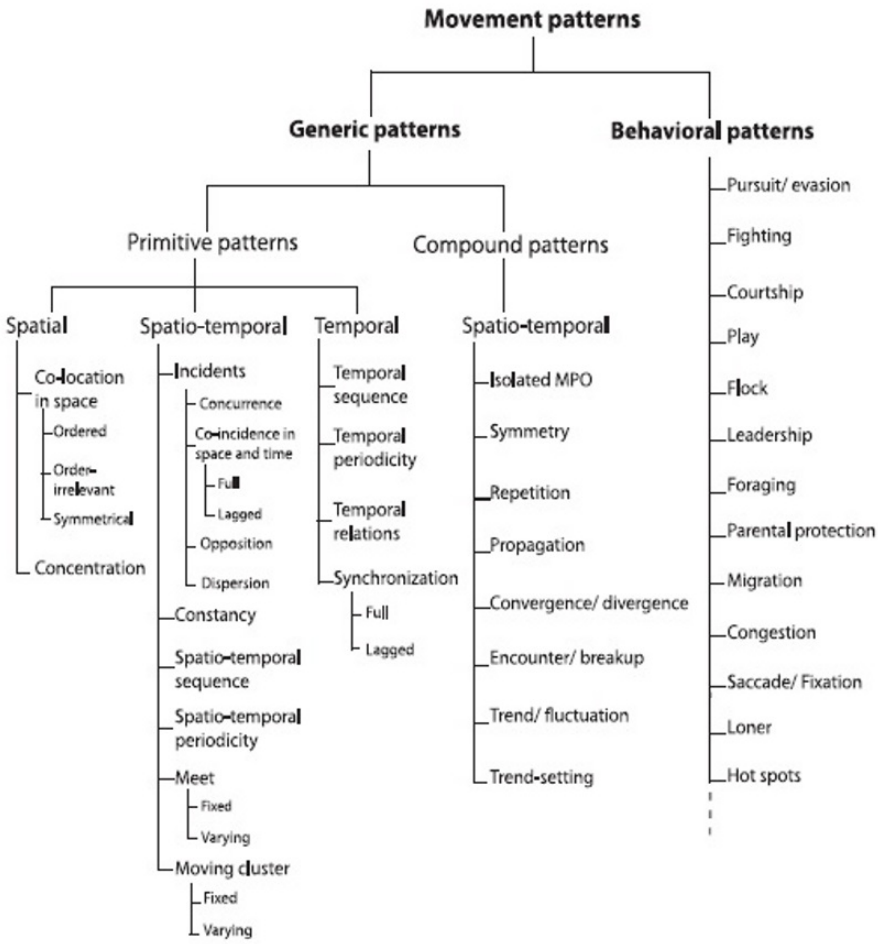


Figure 2.12: Classification of Movement Parameters (as defined in [1])

pattern can be a specific property of a particular kind of *MPO*s. By combining different types of generic patterns, one can construct a behavioural pattern. Generic patterns are further subdivided into primitive patterns and compound patterns. Primitive patterns are the most basic forms of movement patterns, where only a single movement parameter varies. Compound patterns are made up of several primitives involving complex inter-object relations. Primitive patterns can be analysed along three dimensions, namely, spatial, temporal and spatio-temporal dimension. Along the spatial dimension, only spatial location of the object(s) is important for expressing the movement pattern. The temporal dimension is concerned with specific points in time or a duration of time expressible by a temporal interval. Both spatial location and temporal information about these locations are important along the spatio-temporal dimension.

Compound movement patterns analysed along the spatio-temporal dimension only. In Table 2.12, we cite the classification from the work of Dodge et. al. A description of the various types of patterns is given below.

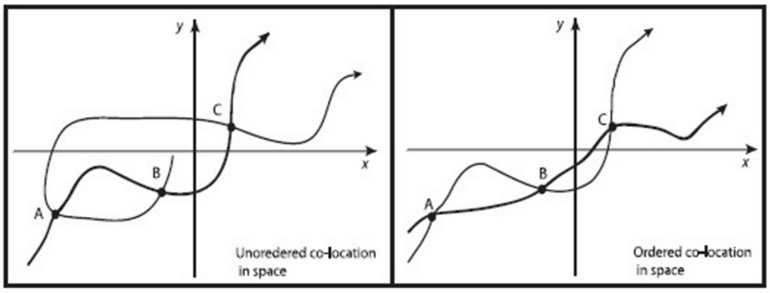


Figure 2.13: Colocation in Space (taken from [1])

**Co-location in space:** In such a pattern, the trajectories of moving objects have some positions in common. In ordered colocation, these common points are arrived at by different entities in the same order whereas in the unordered case it is not so. In symmetrical colocation pattern, the common points are attained in the opposite order [1]. Visit of a set of places by different tourists in the same order may be an example of colocation pattern. Unordered and ordered colocation in space is illustrated in the Figure 2.13. In both left and right part of the figure, two trajectories are shown in an external (allocentric) frame of reference. The points $A$, $B$ and $C$ are common locations in both the trajectories. In the left part of the figure, these points are arrived at in different orders by the objects. One object arrives at these locations in the order $A$, $B$ and $C$ whereas the other object reaches in the order $B$, $A$ and $C$. Therefore, it is an example of unordered colocation in space. The right part of the figure shows an ordered colocation among the objects.

**Concentration:** It expresses spatial concentration of moving objects at a certain instance of time [1]. Congestion of vehicles at a particular zone in the traffic network may be an example. In Figure 2.14, three spatial concentrations of objects in an external frame of reference are shown. This is a snapshot of the locations of the objects at a particular time point $t_i$.

**Incidents:** Occur among multiple object and can be further classified as:

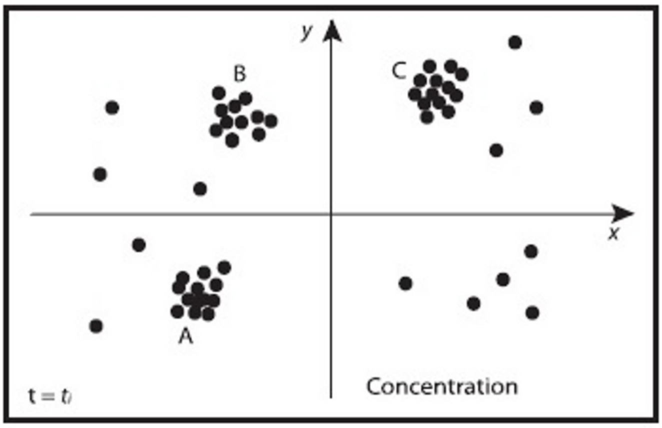


Figure 2.14: Concentration (taken from [1])

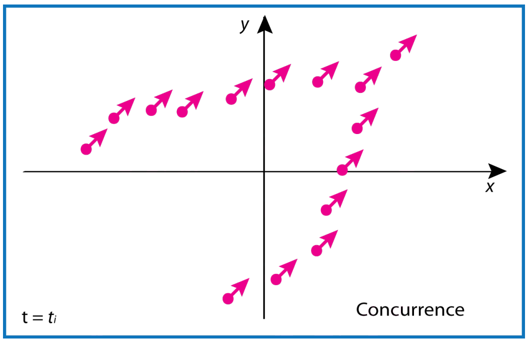


Figure 2.15: Concurrence (taken from [1])

**Concurrence:** It is an incident where a set of entities show the same values of motion attributes at a certain instant or duration e.g. a flock of wild geese flying with similar motion azimuth [1]. In Figure 2.15, directions of movement of a set of MPOs are shown. All these objects are directed in a direction that take them from the left bottom corner to the top right corner.

**Co-incidence in space and time:** It is an incidence that considers similar positions of moving objects. This pattern may be full or lagged [1]. For instance, two different flocks of wild geese reach a particular pond at the same time or separated by a delay of one day.

**Opposition:** Spatial splitting of a group of moving objects shown in a sudden appearance of two opposite motion directions [1]. For instance, when flying geese are prompted to fly in opposite directions by a source of disturbance.

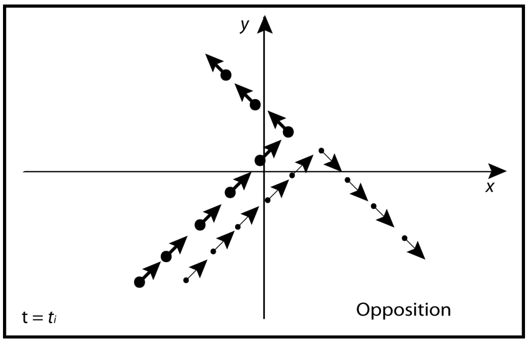


Figure 2.16: Opposition (taken from [1])

In Figure 2.16, at time point $t_i$, the two groups of objects split in two different directions.

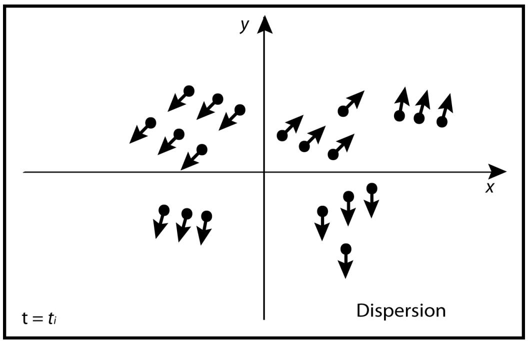


Figure 2.17: Dispersion (taken from [1])

**Dispersion:** It is the opposite of concurrence. An evident pattern in a group of *MPO*s that is performing a non-uniform or random motion [1].In Figure 2.17, it is seen that there is no uniformity in the direction of movement for the objects. The objects are randomly directed along different directions. Though certain objects may move in the same or similar direction, there is no concurrence of motion for the group as a whole.

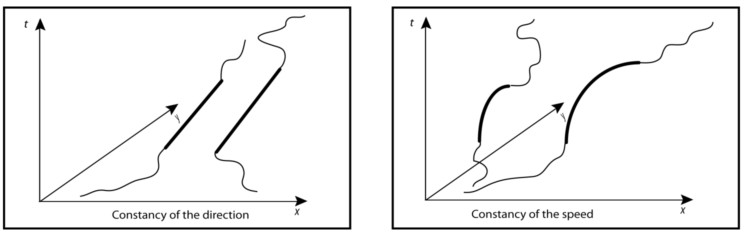


Figure 2.18: Constancy of Direction and Constancy of Speed (taken from [1])

**Constancy:** When the movement parameters remain the same or change insignificantly for a particular duration, e.g. , when a convoy of cars moves along a straight road, at a constant speed, speed and direction and the derived parameters remain the same [1]. In left part of the Figure 2.18, two objects move along the same direction for a time interval. In the right part of the same figure, constancy of speed is illustrated.

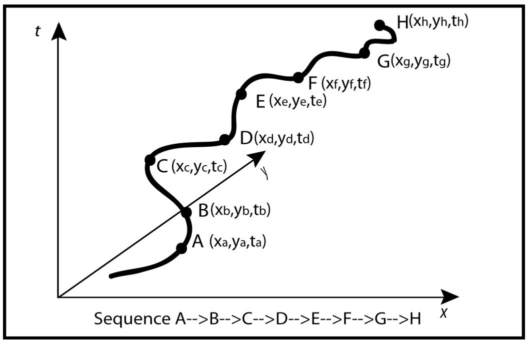


Figure 2.19: Sequence (taken from [1])

**Sequence:** A sequence is an ordered list of visits to a series of locations. It consists of a contiguous series of segments with a known start and end point in space and time. A spatio-temporal sequence refers to an ordered subsequence of locations with their time stamps [1]. As an example of sequential patterns, the tendency of tourists to visit a set of places A to C in a particular sequence ABC within specified duration could be mentioned. In Figure 2.19, the locations from $A$ to $H$ are visited by an object in the order $A$,$B$,$C$,$D$,$E$,$F$,$G$,$H$. Each location has an X-coordinate and a Y-coordinate and each location point is annotated with temporal information.

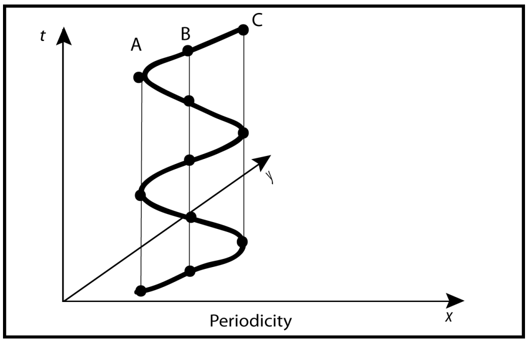


Figure 2.20: Periodicity (taken from [1])

**Periodicity:** A pattern repeats after a definite time duration. The interval of repetition may be daily, weekly, monthly or yearly. In the Figure 2.20, the pattern is visit to three places in the sequence $A$, $B$ and $C$. Along the temporal dimension, it can be seen that these patterns repeat after a fixed duration of time.

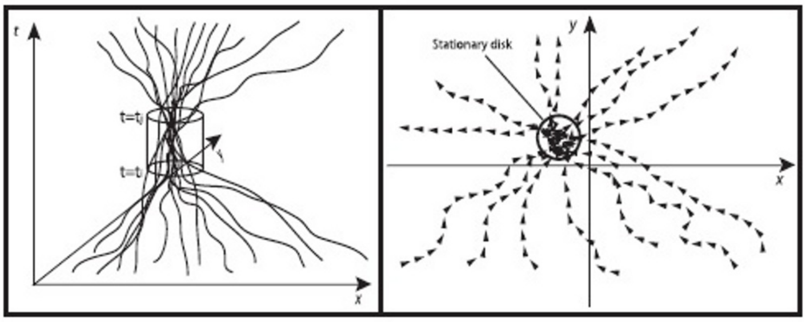


Figure 2.21: Meet (taken from [1])

**Meet:** A meet pattern consists of a set of *MPO*s that form a stationary cluster i.e. they stay within a cylinder of a certain radius in the space-time cube; in the projection to the plane, they stay within a stationary disk of specific radius in a certain time interval [1]. There are two variants of meet, fixed meet and varying meet depending on whether the objects that stay together for a certain duration are the same or change in the meeting region . As an example for a fixed meet pattern, we mention families of geese that gather in the fall in a specific place to form a flock. An example for a varying meet is the rental car drop-off at an airport. In Figure 2.21, the left part shows that moving objects *meet* in a space-time cube for time interval from $t_i$ to $t_j$. The right part of the figure shows the same thing in terms of the projection to the plane. Here, the objects remain within a stationary disk of a specific radius for a certain time duration and this proximity implies their meeting during that duration.

**Moving cluster:** A moving cluster consists of a set of objects that stay close to each other while taking the same path for a specific duration [1]. Nevertheless, it is not necessary that the objects participating in the pattern remain the same, but they may enter and leave, while the cluster is moving. A flock of migrating geese, a convoy of cars following the same route, and troops that move parallel on a military battlefield are different examples of moving clusters. In Figure 2.22,

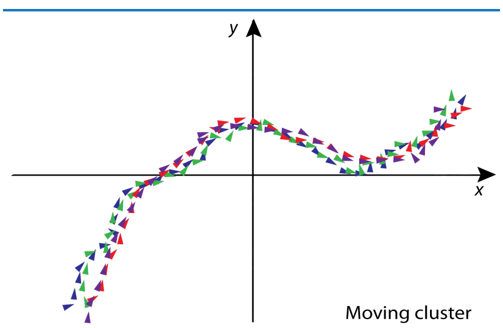


Figure 2.22: Moving Cluster (taken from [1])

four objects are very close together. The direction of movement of these objects are not changing synchronously, but the changes are such that the objects are taking a similar path. This similarity of path is expressed by similarity of changes in movement direction.

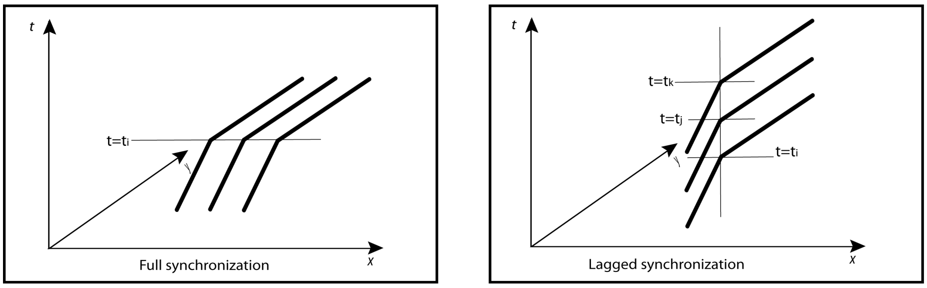


Figure 2.23: Synchronization in Time (taken from [1])

**Synchronization in time:** There are two variants of synchronization patterns. Full synchronization happens when similar changes of movement parameters (e.g., speed, acceleration, direction, etc.) occur at the same time. In contrast, lagged synchronization happens when the changes of movement parameters occur after a time delay [1] e.g. forwards in football (soccer) start moving in a similar direction synchronously, when their goalkeeper kicks the ball towards the opponents side. Full and lagged synchronization are shown in Figure 2.23.

Generic compound patterns are built from primitive patterns. A description of these patterns is given below.

**Isolated object:** An isolated object is a moving object (normally belonging to a group of *MPO*s) that pursues its own path, without any influence on or by the movement of other objects , e.g., when a goose misses the flock and travels

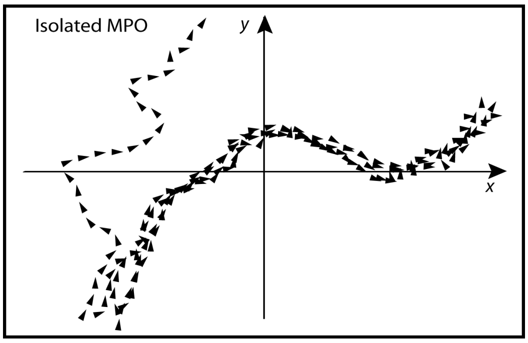


Figure 2.24: Isolated Moving Point Object (taken from [1])

alone [1]. Figure 2.24 shows such an individual MPO that was part of a group for some time. After that it separated out and followed its own course of motion.

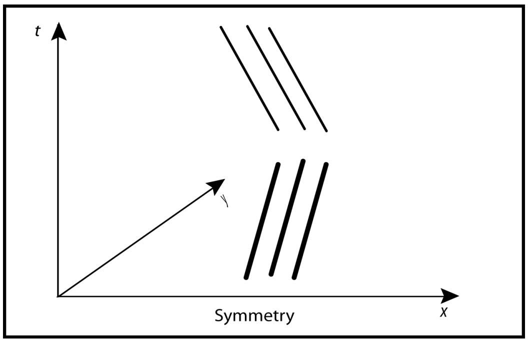


Figure 2.25: Symmetry (taken from [1])

**Symmetry:** Symmetry (shown in Figure 2.25 ) refers to sequences of patterns, where the same patterns are arranged in reverse order, such as wild geese heading north in the spring, and south in the fall.

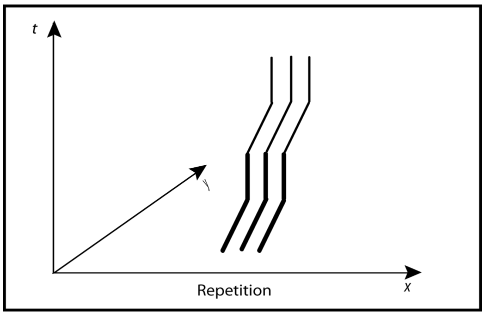


Figure 2.26: Repetition (taken from [1])

**Repetition:** Refers to the occurrence of the same patterns or pattern sequence at different time intervals [1]. For instance, in a football match the wingers may repeatedly sprint along the sidelines or in an eye tracking experiment the test subjects may repeatedly scan the underlying image up and down. Figure 2.26 shows how pattern of movement of three objects alternate from one type to the other.

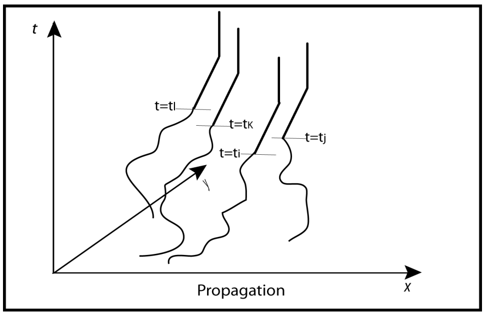


Figure 2.27: Propagation (taken from [1])

**Propagation:** Propagation occurs when one object starts to show a certain movement parameter value, and little by little other objects start adopting the same pattern [1]. By the same token, with every time step more objects are involved. For instance, in the spring snow geese gradually start leaving at different times, depending on how far north they are migrating. The difference to the trend-setting pattern discussed below is that propagation always happens gradually and does not necessarily involve the influence of a trend-setter object. Figure 2.27 shows that an object exhibits a motion pattern starting at time instant $t_i$. Other objects follow this pattern, starting at time instants $t_j$, $t_k$ and $t_l$.

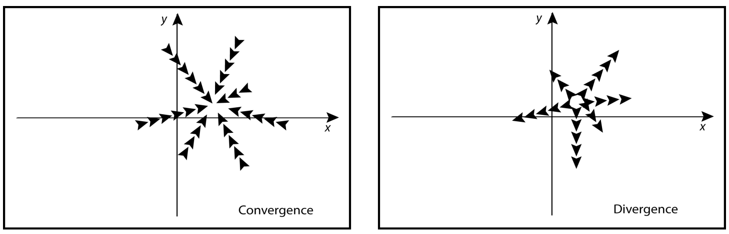


Figure 2.28: Convergence and Divergence (taken from [1])

**Convergence and divergence:** Convergence (shown in left part of the Figure 2.28 ) refers to the movement of a set of objects to the same location, while the original movement direction of the involved objects does not change. In other words, the motion azimuth vectors of the objects involved will be intersecting within a specific range and within a specific duration [1]. The objects need not arrive at exactly the same time. For example, several flocks of snow geese may converge toward a lake to rest. Divergence (shown in right part of the Figure 2.28 ) is defined as the opposite pattern of convergence and describes a group of moving objects that disperse from a common location.

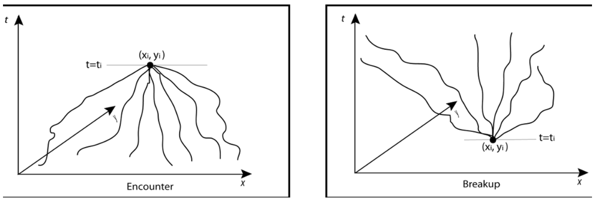


Figure 2.29: Encounter and Breakup (taken from [1])

**Encounter and breakup:** Encounter (shown in left part of the Figure 2.29 ) refers to moving to and meeting at the same location. Encounter is a specific form of convergence pattern where the objects arrive at the same time. In an encounter pattern a set of *MPO*s have motion azimuth vectors that can be extrapolated from the current movement such that the vectors intersect within a specific range and the *MPO*s meet at the same time [1]. Breakup (shown in right part of the Figure 2.29 )is defined as the opposite of the encounter pattern and describes a divergence, adding a temporal (concurrency) constraint. In a football match, an encounter occurs when several players rush towards the ball and reach it at the same time. A breakup occurs when ducks flee from a pond after a gunshot is heard.

**Trend and fluctuation:** Trend refers to consistent changes in the movement parameters of moving objects . e.g., for an airplane circling in a holding pattern the rate of change of the movement direction will remain constant [1]. Conversely, fluctuation refers to irregular changes in the movement parameters of moving objects , e.g., a flock of geese may change their flying formation between V-shape, irregular V-shape, or sometimes lines. In left part of Figure 2.30,

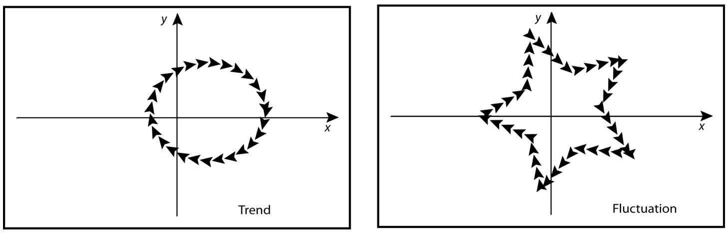


Figure 2.30: Trend and Fluctuation (taken from [1])

the direction of movement of an object changes consistently forming a circular shape. In the fluctuation pattern, shown to the right, the direction changes in such a way that a fluctuation between high and low points can be observed.

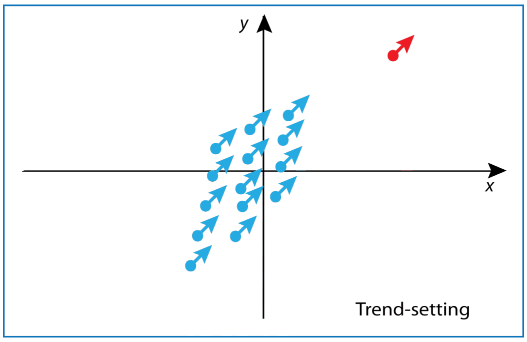


Figure 2.31: Trendsetting Pattern (taken from [1])

**Trend-setting:** Trend-setters are defined as objects that anticipate a certain movement pattern that is afterwards followed by a subset of the other moving objects [1]. In another words, trend-setters are objects that influence the movement of others not necessarily in a spatial and temporal proximity. For example, in a football match, a striker executing a sudden rush towards the adversary goal acts as a trend-setter, anticipating (or triggering) a similar movement direction by the defenders and his/her own teammates). There are two variants of trend-setting, non-varying trend-setting with a fixed subset of followers and varying trend-setting. In the case of varying trend-setting, the subset of followers may change over the time intervals of the observation duration. Similarly to a moving cluster, in the trend-setting pattern objects move in the same direction or may have other similar movement characteristics such as same speed or acceleration. A trendsetting pattern can be observed in Figure 2.31. The trend setter moves in a particular direction and other objects follow in the same direction.

Literature survey reveals that statistical approaches are predominantly used in the filed of computer vision. These approaches are not adequate if we need to represent internal structure of a motion pattern. Grammar based approaches are also proposed, but representation of concurrency is a problem in these approaches. In GIScience, majority of approaches use ad hoc algorithms for extracting patterns. This approach is not general in the sense that it does not provide a general framework for recognition of motion patterns.

In the next chapter, we introduce two qualitative formalisms, one for direction and the other for spatial orientation. We have used these formalisms in learning and representation of a motion pattern between two persons from video data. The formalisms are introduced early because these formalisms have been used in examples in subsequent chapters.

## 2.4 Syntactic Pattern Recognition

Statistical pattern recognition attempts to classify patterns based on a set of extracted features. An underlying statistical model is used for the generation of these patterns. A numerical representation of the features of the pattern in the form of a vector is used as a representational technique. Statistical approaches are widely used for their simplicity. A multidimensional feature vector represents the patterns to be recognised. Each pattern is represented by a point in this multidimensional feature vector space. Distances between such points are measured in statistical space for recognition of patterns. Statistical approaches demand determination of the feature vector along with training and classification of patterns.

Patterns may sometimes contain structural and relational information that are difficult to quantify in feature vector form. Syntactic pattern recognition is a form of pattern recognition that allows us to analyse structural similarity of patterns. Description of pattern structure is advantageous in situations where a simple classification is not possible [49]. Moreover, it is possible to describe aspects that cause a pattern to not be assigned to a particular class. In certain applications, recognition of a pattern is possible only through a description of its structure. As an example, we can cite the case of picture recognition and scene analysis. These applications have a large number of features. The structure of the pattern is complex. Therefore, a hierarchical description of these complex patterns in terms of simple primitives is necessary. An analogy can be drawn between this case and the syntax definition of formal languages. In a formal language, tokens are built from character sets and statements are built hierarchically from these tokens. This is similar to hierarchical construction of patterns from sub-patterns and primitives [50]. Syntactic pattern recognition, therefore, uses formal grammars for representation and recognition of patterns [51], [52]. The simplest sub-patterns are known as *pattern primitives.* These *pattern primitives* are recognised directly in input data stream. A *pattern description language* is used to define the structure of patterns in terms of *pattern primitives.* In syntactic recognition, a pattern is thus represented as a string of primitives. Pattern recognition is equivalent to parsing the structure of such a string according to the formal grammar used for defining the pattern [53]. This approach is advantageous in the sense that an infinite number of patterns can be represented in compact way using a formal grammar [54]. This type of pattern recognition can take handle more complex relations between features than numerical feature vectors used in statistical recognition. The challenge of grammar based pattern recognition is to propose grammars that have decidable and efficient algorithms for parsing [50].

In a general sense, a formal language is defined as a set of strings over some alphabet $\Sigma$. The alphabet is a finite non-empty set of symbols. A string over an alphabet $\Sigma$ is a finite sequence of symbols of $\Sigma$. We can represent the rules that characterise the strings of a language using a grammar [55]. Grammars represent the most general system of representing languages.

*A grammar is a quadruple (*$\Sigma$, V, S, P*), where:*

1. $\Sigma$ is a finite nonempty set called the terminal alphabet. The elements of $\Sigma$ are called the terminals.

2. $V$ is a finite nonempty set disjoint from $\Sigma$. The elements of V are called

the non-terminals or variables.

3. $S \in$ V is a distinguished non-terminal called the start symbol.

4. $P$ is a finite set of productions rules of the form $\alpha \rightarrow \beta$ where $\alpha \in (\ \Sigma \cup V\ )^*\ V\ (\ \Sigma \cup V\ )^*$ and $\beta \in (\ \Sigma \cup V\ )^*$.

Chomsky provided a classification of grammars into four classes [56]. This classification is obtained by imposing stricter restrictions on the forms of productions. In *Type-0* or unrestricted grammar, no restriction is imposed on the grammar productions. In a *Type-1* or context-sensitive grammar each production is of the form $\alpha A\beta \rightarrow \alpha\gamma\beta$ where $\gamma \neq \epsilon$. In *Type-2* or context free grammar, in any production of the form $\alpha \rightarrow \beta$, $\alpha$ is a single non-terminal. A grammar is called a *Type-3* or regular grammar if each production has one of the forms: $A \rightarrow c\ B$, $A \rightarrow c$ and $A \rightarrow \epsilon$. Language generated by *Type-1* grammar is known as context sensitive language. *Type-2* grammars generate context free languages and *Type-3* grammars generate regular languages. The language generated by *Type-0* grammars is recognised by *Turing machines*. *Linear bounded automata* are the recognisers for context sensitive languages. Context free languages are recognised by *pushdown automata* whereas *finite automata* recognise regular languages.

Given these four classes of languages, an important problem is the membership problem. Membership problem can be stated as: Given a string over $\Sigma$, does this string belong to *L(G)*? For *Type-0* grammars, the membership problem is undecidable in general. For *Type-1* grammars this problem is *PSPACE-Complete*. For context free languages, this problem is decidable in polynomial time and for regular languages it is decidable in linear time [57].

In the work presented in this thesis, patterns are represented as strings of terminals. These terminals are derived from a qualitative description of a pattern. Therefore, learning of a regular grammar (equivalently a DFA) is necessary. So,it will be relevant to discuss some approaches in learning DFA. Learning a deterministic finite automaton from both positive and negative data is a well-known problem in grammatical inference. Gold established that the associated combinatorial problem is not tractable [58]. Pitt and Warmuth [59] showed that finding a polynomially larger DFA than the minimum DFA, consistent with data, is NP-Hard. Gold [58] gave an algorithm that works when data are sufficient; but can not generalise if this is not the case. Trakhtenbrot and Barzdin proved that in the case where all the data are presented upto a certain length, it is possible to learn a DFA. The problem here is that the volume of data becomes huge. Dupont et. al. [60] proposed a lattice based approach for learning a DFA where the number of nodes in the lattice grows exponentially. Lang [61] showed experimentally that when volume of data increases beyond a limit, learning becomes very poor. After this different approaches have been applied to solve this problem. Notable work include evidence driven technique (Lang [60]), data driven heuristics ( [60]), TABU search by Giordano [60], genetic algorithm by Dupont [60], incremental algorithm by Parekh and Honavar [60], learning of a non-deterministic automaton first by Denis et. al. [60], learning non-deterministic automaton from queries and counter-examples by Yokomori [62], algorithms by Oliveira [63] and Lang [64] on learning large automata etc.

In this thesis, we do not need to consider solving the general DFA learning problem from data. Rather, we need to handle a very special subclass and therefore learning algorithm becomes simple. The notion of spatio-temporal continuity, found in QSR represented events, makes the pattern strings structurally very simple. In Chapter 4, we have shown that by exploiting spatio-temporal continuity, it is possible to represent pattern strings in a form where a terminal does get repeated consecutively inside the string. Therefore, a sequence of non-repeated terminals can be represented by a set of simple production rules. Whenever a new example is seen, we need to check whether the corresponding string representation occurs as a substring of already learned strings. If not, this new string needs to be learned and represented using a set of productions. In our work, only positive examples are learned and represented.

## 2.5 Motion Pattern Analysis in Computer Vision

Activity recognition from video has been a significant research area. The reason is that it finds application in a diversified set of areas like security and surveillance, content based video retrieval, animation, behavioural biometrics, sports analysis etc. Activity recognition is closely related with motion pattern recognition. Analysis of movement of participating entities is a central focus in activity recognition problem. Activity recognition is hierarchical in nature. This hierarchy is in terms of a set of modules that are organised hierarchically. Low level modules perform tasks such as background-foreground segmentation, tracking and object detection. The middle level is concerned with action recognition. An action occupies a short span of time. At the highest level, semantics of the activity are derived from the low level actions.

Within the computer vision community, activity recognition focuses mainly on human activity recognition. A common and widely used way to model the structure of human behaviors relies on purely probabilistic approaches. *Hidden Markov Models (HMM)* is one such widely used model. The general idea of these approaches is to extract sets of features from the low-level data and feed them into the probabilistic graphical model used to define the event structure. Yamato et. al. [65] used *HMM*s to recognize tennis shots such as backhand stroke, backhand volley, forehand stroke, forehand volley, smash etc. Successful gesture recognition systems using *HMM*s have been reported in [66], [67] and [68]. *HMM*s have also been used in modeling the temporal evolution of human gait patterns for action recognition and biometrics in [69], [70] and [71]. Brand et. al. [72] proposed a coupled *HMM* to represent the dynamics of interacting objects [72]. Moore et. al. [73] used *HMM*s combined with object detection modules to analyse the relationship between actions and objects [73]. In the work of Hongeng and Nevatia [74], a priori beliefs of state-duration were incorporated into *HMM* framework. The resultant is called *Hidden semi-Markov Model.* Cuntoor and Chellappa [75] proposed a mixed-state *HMM* formalism to model non-stationary activities. A special type of *HMM* (Switching Hidden Semi-HMM) with a two layer implementation is reported in the work of [76] to learn and recognise human activities. Nguyen et. al. [77] proposed a hierarchical *HMM* for recognising the hierarchical structure and the shared semantics contained in the movement trajectories. A scalable method for complex activity recognition is presented in [78].

*Linear dynamical systems (LDS)* are a more general form of *HMM*s where the state-space is not constrained to be a finite set of symbols but can take on continuous values. The *LDS* can be interpreted as a continuous state-space generalization of *HMM*s with a Gaussian observation model. Several applications such as recognition of humans and actions based on gait [79] , [80], activity recognition [75] and dynamic texture modeling and recognition [81] , [82] have been proposed using *LDS*s. Advances in system identification theory for learning LDS model parameters from data [83], [84], [85] and distance metrics on the LDS space [86], [81], [87] have made LDSs popular for learning and recognition of high-dimensional time series data. In-depth study of the LDS space has enabled the application of machine learning tools on that space such as dynamic boosting [88], kernel methods [89], [90] and statistical modeling [3]. Newer methods to learn the model parameters [91] have made learning much more efficient than in the case of *HMM*s.

Bregler [92] presented a multilayered approach to recognize complex movements consisting of several levels of abstraction. North et. al. [93] augment the continuous state vector with a discrete state component to form a *mixed* state. Pavlovic and Rehg [94], [95] model the non-linearity in human motion in a similar framework, where the dynamics are modeled using *LDS* and the switching process is modeled using a probabilistic finite state-machine. Though the *SLDS* framework has greater modeling and descriptive power than *HMM*s and *LDS*s, learning and inference in *SLDS* are more complicated, often requiring approximate methods [96]. In practice, determining the appropriate number of switching states is challenging and often requires large amounts of training data or extensive hand tuning.

A *Bayesian network (BN)* [97] is a graphical model that encodes complex conditional dependencies between a set of random variables which are encoded as local conditional probability densities (CPD). *Dynamic Belief networks (DBNs)* are a generalization of the simpler Bayesian networks by incorporating temporal dependencies between random variables. *DBN*s encode more complex conditional dependence relations among several random variables as opposed to just one hidden variable as in a traditional *HMM*. Buxton and Gong [98] used Bayesian networks to capture the dependencies between scene layout and low level image measurements for a traffic surveillance application. Remagnino et. al. [99] present an approach using *DBN*s for scene description at two levels of abstraction. Modeling two-person interactions such as pointing, punching, pushing, hugging etc. was proposed by Park and Aggarwal [100] in a two-stage process. Intille and Bobick [101] use Bayesian networks for multiagent interactions where the network structure is automatically generated from the temporal structure provided by a user. Usually the structure of the *DBN* is provided by a domain expert. But this is difficult in real life systems where there are a very large number of variables with complex inter-dependencies. To address this issue Gong et. al. [102] presented a *DBN* framework where the structure of the network is discovered automatically using Bayesian Information Criterion [103] , [104]. *DBN*s have also been used to recognize actions using the contextual information of the objects involved. Moore et. al. [73] conduct action recognition using belief networks based on scene context derived from other objects in the scene. Gupta et al. [105] present a Bayesian network for interpretation of human-object interactions that integrates information from perceptual tasks such as human motion analysis, manipulable object detection and *object reaction* determination. Use of Bayesian networks in anomaly detection of vessel tracks is reported in the work of Steven et. al. [106]. Phan has proposed an automatic decision tree pruning method for improving activity recognition [107]. A survey of human activity interpretation in video sequence can be found in [108]. In another work of Borges et. al. [109], a survey of video-based human behaviour understanding can be found. Ontologies are used for recognition of human behaviour. Natalia et. al. [110] have provided a survey on the ontological issues of human behaviour recognition. Holistic and posed based methods for activity recognition are proposed in [111].

Works highlighting the importance of qualitative spatial relations for video understanding have been reported in literature. Early attempts in this area includes the work of Fernyhough et. al. who have shown how qualitative relations can be automatically learned from a video input [112]. Qualitative spatio-temporal relations have been used with variable length Markov model to learn interaction between objects in a scene [113]. Qualitative spatio-temporal relations between objects have been reported in the work of Sridhar [12] for unsupervised learning of event classes from video. Relational representation of scenes using qualitative spatial relations is reported in the work of Dubba et. al. [11]. An approach to integrate spatial aspects like topology, direction, size and distance of moving objects is proposed in the work of Cohn et. al. [114]. Sadeghi et. al. have proposed a modified version for extracting spatial information with much higher accuracy [115]. A qualitative spatial representation of *General Solid Rectangle*s is represented in [116]. In [117], combination of topological and directional information for two dimensional spatial objects is studied. Use of qualitative spatial reasoning in improving tracking accuracy in reported in [118].

Using formal grammars, we can represent the structure of an activity. Use of formal grammars is advantageous because it permits principled inference and provably correct analysis. Moreover, efficient parsing algorithms exist for certain classes of formal grammars [119], [120]. Here, efficiency refers to time complexity of parsing algorithms. For example, for regular languages, parsing can be done in linear time and for context free languages, it can be done in polynomial time. An early work on the use of grammars for activity recognition is reported in [121]. In this work, a deterministic grammar with no probabilistic modelling is used for recognising hand manipulations in sequences containing disassembly tasks. Ryoo and Aggarwal [13] used the context-free grammar (CFG) formalism to model and recognize composite human activities and multiperson interactions. In their work, *HMM*s were used at low-levels and higher level semantics are modeled by *CFG*s. Algorithms for detection of low-level primitives are frequently probabilistic in nature. *Stochastic Context-free grammar (SCFG)* was used by Ivanov and Bobick to represent semantics of activities [14]. Structure of these activities was assumed to be known a priori. In their work, *HMM*s were used for detection of low-level primitives. Probabilities were used along with grammar productions. Use of *skip* transitions made the system robust against errors. Moore et. al. [122] used *SCFG*s to model multi-tasked activities. Probabilistic attribute grammars have been used by Joo and Chellappa [123] for multiagent activities in visual surveillance. A computational framework is proposed in [124] for recognising behaviour in a minimally supervised manner. Zhang et. al. [125] have proposed to extract the terminal symbols of a *SCFG* from motion trajectories. Motion trajectories are transformed in a set of basic motion patterns (primitives) that are taken as terminals for the formal grammar. A rule induction algorithm, based on the Minimum Description Length (MDL), automatically extracts the spatio-temporal structure of the event from the primitive stream. Temporal logic between atomic events is modeled through a combination of *SCFG* and Allens temporal logic [18]. A Multi-Thread Parsing algorithm with Viterbi-like error recovering is developed in order to recognize events in the stream. A method for learning a context-free data automatically from input motion data is reported in [126]. A *Gesture Description Language* is proposed in [127] for recognition of human body poses and gestures in real time. A context-free grammar, named as *Manipulation Action Grammar*, is proposed by Yezhou et. al. [128] for understanding human manipulation actions. A hierarchical recognition strategy using support vector machine, *HMM* and formal grammars is used for real-time 3D motion recognition in [129]. Heryadi et. al. [53] have proposed a method to recognise basic 3D dance motions using a stochastic regular grammar from training data set. Recognition of human behaviour using a context-free grammar is proposed in the work of Andrea et. al. [51]. A *Featurebased Stochastic Context-Free Grammar* is used in [130] for learning and recognising natural hand gestures. A review of vision-based human action recognition is presented in [131].

Inductive logic programming (ILP) is a subfield of machine learning which uses logic programming as a uniform representation for examples, background knowledge and hypotheses. When an encoding of the known background knowledge along with a set of examples is represented as a logical database of facts, an ILP system will derive a hypothesised logic program which entails all the positive and none of the negative examples. ILP has been explored as a way of learning events from video. A notable work on learning QSR represented events from complex videos within an ILP based framework is reported in [11] . In this work, a supervised ILP based framework was applied on a large (approximately 2.5 million frames) and noisy video dataset from airport apron to learn events. In this approach, a type refinement operator is used to reduce the number of false positives and learn semantically meaningful hypotheses.