Chapter 5
Trajectory Pattern Mining

Hoyoung Jeung, Man Lung Yiu, and Christian S. Jensen

Abstract :
In step with the rapidly growing volumes of available moving-object trajectory data, there is also an increasing need for techniques that enable the analysis of trajectories. Such functionality may benefit a range of application area and services, including transportation, the sciences, sports, and prediction-based and social services, to name but a few. The chapter first provides an overview trajectory patterns and a categorization of trajectory patterns from the literature. Next, it examines relative motion patterns, which serve as fundamental background for the chapter's subsequent discussions. Relative patterns enable the specification of patterns to be identified in the data that refer to the relationships of motion attributes among moving objects. The chapter then studies disc-based and density-based patterns, which address some of the limitations of relative motion patterns. The chapter also reviews indexing structures and algorithms for trajectory pattern mining.

5.1 Introduction

We are witnessing a rapid and continued diffusion of mobile devices such as smartphones, personal navigation devices and tablet computers. Further, these devices are increasingly being geo-positioned using satellite navigation systems, e.g., GPS, systems that exploit the wireless communication infrastructure, and proximity-based systems, e.g., RFID-based systems.

Hoyoung Jeung
École Polytechnique Fédérale de Lausanne, Switzerland, e-mail: hoyoung.jeung@epfl.ch

Man Lung Yiu
Department of Computing, Hong Kong Polytechnic University, Hong Kong, e-mail: csmlyiu@comp.polyu.edu.hk

Christian S. Jensen
Department of Computer Science, Aarhus University, Denmark, e-mail: csj@cs.au.dk

The resulting *location-aware* devices find widespread use in various business and personal settings in society. As a consequence of this development, increasing volumes of position data are being accumulated, and the capability of analyzing large volumes of trajectory data is in increasing demand in a wide spectrum of applications.

Significant applications in various domains need to identify and utilize groups of trajectories that exhibit similar patterns from a collection of trajectories. Example applications include transportation optimization, prediction-enabled services, scientific and social analysis applications, sports analyses, as well as crowd and outlier analyses.

- Transportation optimization applications [25] need to find groups of similar trajectories that indicates that the corresponding objects traveled together. For instance, a car pooling application may connect drivers in the same trajectory group in order to reduce their travel expenses. A logistics application may examine the delivery trucks in the same group in order to achieve better planning.
- Prediction methods [50] may exploit knowledge of trajectory groups for the understanding of object behavior. Such knowledge can be used for offering effective notifications, for the delivery of advertisements to targeted audience, and for providing customized location-based services.
- Scientific studies may call for the identification of groups of animals that moved together. They are useful in discovering animal movement patterns (e.g., bumble bees, a variety of birds, sea turtles, whales, and fish) [2], in finding herds of animals, and in studying animal behavior patterns in habitats.
  Similarly, social analysis studies may aim to identify socio-economic patterns [23] from typical movement patterns of individuals.
- Team sports events [30, 1] (soccer, baseball, hockey, rugby, digital battle fields) also provide valuable trajectory data that capture the players' movements. By studying a game as groups of trajectories, it may be possible to better understand the game [30], to analyze the tactics used in the game, and to even extract the location and time of using a certain strategy.
- Traffic analysis applications may utilize trajectory groups for the study of crowds and outliers. In this scenario, a moving object can be either a vehicles on the road or a pedestrian on the street. A large trajectory group is likely to indicate a crowd behavior. By identifying crowds from the trajectories, a better understanding of crowds is possible, e.g., the times and places when and where crowds form and dissolve. Such information may be exploited for managing transportation infrastructures effectively.
  It is also of interest to mine outliers, which do not belong to any trajectory group. This may be used for detecting and removing errors in the trajectory data (e.g., finding a device with a malfunctioning GPS receiver). It may also be applied for identifying dangerous driving behaviors.

Trajectory pattern mining is an emerging and rapidly developing topic in the areas of data mining and query processing that aims at discovering groups of trajectories based on their proximity in either a spatial or a spatiotemporal sense. The literature contains a variety of recent proposals in this area.

Existing proposals represent different trade-offs in the expressiveness of the trajectory patterns studied. Considering only restricted patterns may result in not being able to identify interesting phenomena from the data, whereas considering quite relaxed patterns may lead to the reporting of insignificant patterns.

Existing proposals also come with their own index structures and mining algorithms that aim to enable efficient and scalable discovery of patterns in large trajectory datasets. This chapter presents an overview of the key concepts and discovery techniques in state-of-the-art studies in the mining of trajectory patterns.

The rest of the chapter is organized as follows. Section 5.2 introduces the concept of trajectory pattern and provides a categorization of patterns. We then study relative motion patterns in Section 5.3, presenting disc-based patterns in Section 5.4, and examining density-based patterns in Section 5.5. Section 5.6 covers distance measures and methods for mining trajectory patterns. We summarize the chapter in Section 5.7.

5.2 Overview of Trajectory Patterns

5.2.1 Pattern Discovery Process

A trajectory of a moving object is a continuous function from the time domain to the domain in which the movement occurs. In animal tracking, the objects typically move freely in the water, on the surface of the Earth, or in the air. In such cases, the movement domain is often modeled as two- or three-dimensional Euclidean space. In settings where the objects are vehicle, the movement domain is often modeled as a spatial graph that models a road network. Further, the spatial extent of a moving object is typically ignored so that the position of an object at a given time is modeled as a point.

Figure 5.1 illustrates the architectural context for managing trajectories. Multiple objects, each with a mobile device with a unique identifier *id*, contribute data. Each device samples the location of its object according to some policy. For example, devices may take a position sample at a fixed frequency such as every second. A location record has the format (t, x_t, y_t, id), where the sampled location at time t is (x_t, y_t).

The true trajectory of an object is unknown. The location records available for an object are used to create an approximation of the object's true trajectory. Specifically, existing techniques for trajectory pattern mining typically assume that the trajectory of an object is given by a polyline, i.e., a sequence of connected line segments.

A server is employed for storing and managing the collected trajectories from all devices. Important steps in the handling of trajectories are described next.

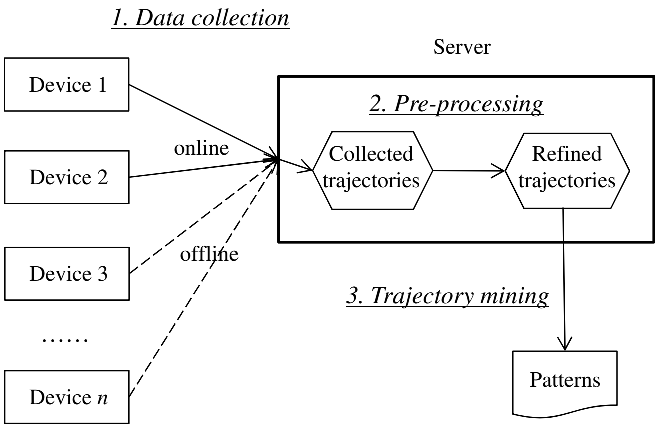


Fig. 5.1 Architecture.

1. *Data collection.*
   In this step, devices submit their location records to the server. Online devices may submit their data in real time or delayed, in batches. For example, a device may submit records as soon they become available on the device, or it may collect a small batch of records before submitting in order to save bandwidth.
   Offline devices accumulate their trajectory data, which is transferred to the server in batches via other means. For example, an offline navigation device in a vehicle may occasionally be connected to an online personal computer in order to transfer location records to the server.
2. *Pre-processing.*
   Due in part to imperfections in the positioning technologies employed and the use of different sampling policies, collected trajectories may be inhomogeneous and may not be of the desired quality. For instance, some of trajectories may lack location records for times where other trajectories posses such records.
   In pre-processing, the server "cleans" the data and converts it into a standard format in preparation for the data mining. As part of the pre-processing, the granularity and the representation of the refined trajectories need to be decided by the server. For example, a finest sampling rate is often chosen, and it is decided whether a trajectory be stored as a sequence of location records or should be approximated by a piecewise linear function from time to space.
3. *Mining trajectories.*
   The refined trajectories are made available for the mining algorithms to discover patterns (e.g., clusters of trajectories).
4. *Post-processing.*
   This step enables applications to analyze or tune the discovery process. For example, mined patterns can be displayed interactively using visualization tools, which can suggest more appropriate parameter values for the mining algorithms or finding further research issues.

5.2.2 *Classification of Trajectory Pattern Concepts and Techniques*

The term *trajectory pattern* covers many different types of patterns that can be mined from trajectory data [29]. As a result, the concepts and techniques underlying trajectory pattern discovery are classified by according to a variety of aspects [19]. In the following, we describe key classifications of trajectory pattern concepts and techniques while emphasizing the intuitions behind the classifications.

5.2.2.1 Mining Tasks on Trajectories

Two typical mining tasks on trajectories are *clustering* and *join*. Clustering aims at discovering groups of similar objects from a single trajectory collection. On the other hand, a join is a specific operation that computes pairs of similar objects from two trajectory collections.

Clustering of Trajectories

Clustering is the process of organizing objects into groups so that the members of a group are similar and so that members of distinct groups are dissimilar, according to some specific definition of similarity. Clustering is a fundamental concept in data mining, due to its wide spectrum of applications.

A clustering technique is classified according to the type of data that it is applicable to. Clustering techniques for different types of data are likely to vary significantly with respect to the underlying concepts and the specific techniques employed.

In the context of trajectory data, clustering attempts to group trajectories based on their geometric proximity in either spatial or spatiotemporal space. Clustering of trajectories is perhaps one of the most fundamental operations used in various types of trajectory pattern mining, since the discovery of trajectory patterns typically involves the process of grouping similar positions, trajectories, and objects.

Assuming the polyline representation of a trajectory, one fundamental approach to trajectory clustering treats a segment of a trajectory as a minimum unit when computing the distance between two trajectories. To this end, it is essential to design an effective distance function for measuring the distance between two trajectories (segment). We offer details on trajectory distance measures in Section 5.6.1. Another fundamental approach views a cluster of trajectories as a sequence of spatial clusters. Assuming that all objects have position samples at the same points in time, such spatial clusters can be obtained by applying some spatial clustering technique to the positions for each point in time.

This chapter describes in detail the core ideas and concepts of state-of-the-art algorithms for clustering of trajectories: *relative motion patterns, flock, TRACLAUS, moving cluster, convoy,* and *swarm*.

Trajectory Join

Some trajectory patterns are defined and computed by means of join queries. Given two data sets P_1 and P_2, spatio-temporal joins find pairs of elements from the two sets that satisfy a given predicate with both spatial and temporal attributes [31, 17]. The study of airplane or vessel trajectories with the objective of finding incidents may be accomplished using joins. Since joins may involve the comparison of all trajectories in data set P_1 with all trajectories in data set P_2, which is computationally expensive, a common approach for join processing involves the use if indexing techniques to avoid unnecessary distance computations. For example, Tao et al. [53] show how join queries are processed by using the time-parameterized methods [56, 52, 51].

The *close-pair join* [13] reports all object pairs (o_1, o_2) from $P_1 \times P_2$ with distance $D_\tau(o_1, o_2) \leq e$ within a time interval τ, where e is a user-specified distance. Plane-sweep techniques [6, 61] have been proposed for evaluating spatio-temporal joins. Zhou et al. [61] use join predicates that define a rectangular region in time and space. An index structure (MTSB-tree) is introduced to enable efficient retrieval of the pairs of trajectories that satisfy the join predicates. Instead of using an index, Arumugam and Jermaine [6] utilize MBR approximations of trajectory segments to reduce the computation of query processing.

As does the close-pair join, the *trajectory join* [10] aims at retrieving all pairs of similar trajectories in two data sets. Bakalov et al. [9, 10] represent trajectories as sequences of symbols, and apply sliding window techniques to measure the symbolic distance between possible pairs.

5.2.2.2 Spatial and Spatiotemporal Patterns

Groups of similar trajectories carry different semantics depending on whether they are grouped according to spatial or spatiotemporal geometric proximity. Figure 5.2 demonstrates the difference between *spatial* and *spatiotemporal* trajectory patterns. At time $t = 1$, two objects o_1 and o_2 start moving closely together in the direction of the upper-right corner. At time $t = 2$, a new object o_3 appears and starts to move from a close location to where o_1 and o_2 were at time $t = 1$. Meanwhile, o_1 and o_2 have reached the center of the movement space and continue to move. At time $t = 3$, o_3 keeps following a path similar to those that o_1 and o_2 have been following, while o_1 and o_2 complete their journeys. Finally, o_3 finishes its journey at time $t = 4$ close to where o_1 and o_2 stopped.

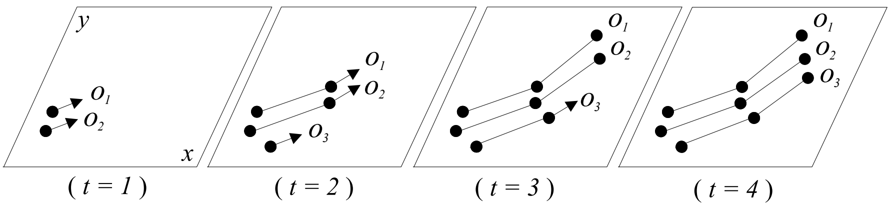


Fig. 5.2 Construction of three objects' trajectories during $t = [1, 4]$.

Given the trajectories of o_1, o_2, and o_3 collected until $t = 4$, the embeddings of the trajectories into the movement space look similar. When we cluster the trajectories according to the similarity of the embeddings, the three trajectories may form one cluster, although o_3 has never traveled together with o_1 and o_2, but has merely followed the same route as these.

The above is an example of a *spatial trajectory pattern* (Figure 5.3(a)). Despite the fact that the concept of trajectory encompasses the time domain, a variety of applications do not need to consider the temporal aspects of trajectories.

For example, analyzing a set of hurricane trajectories collected over several years for forecasting the trajectories of future hurricanes may be done best by considering simply the embeddings into the movement space of the past hurricanes so that is becomes the similarity of the routes of the hurricanes that is studied. Put differently, the routes of the past hurricanes may be the most important aspect when aiming to offer pre-warnings to regions of potential damage.

In contrast to spatial trajectory patterns, *spatiotemporal trajectory patterns* take the time information in a trajectory into account. For example, o_3 in Figure 5.2 may not belong to the same cluster as the one o_1 and o_2 belong to (see Figure 5.3(b)), since o_3 always trailed behind o_1 and o_2. Having a time constraint, a spatiotemporal

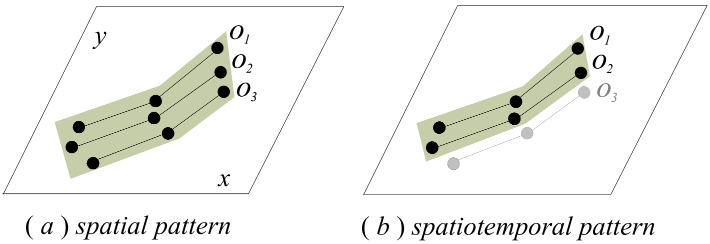


Fig. 5.3 Spatial versus spatiotemporal trajectory patterns.

trajectory pattern is therefore stricter than a spatial trajectory pattern. This time constraint can help discover more specific patterns in many cases of mining trajectory data. For instance, identifying objects that have traveled together for the purpose of given car pooling recommendations should consider the temporal information of trajectories. Discovering places in a road network where congestion may occur based on vehicle trajectories should take into account the time information of the trajectories.

5.2.2.3 Granularity of Trajectory Patterns

The granularity of a trajectory pattern can be characterized by the time interval during which the pattern holds and the number of objects that are involved in the pattern.

Global Vs. Partial Patterns

Trajectory patterns can be classified based on the temporal extent of the trajectories that participate in the patterns. In *global trajectory patterns*, trajectories are viewed as non-decomposable, i.e., the basic unit of pattern discovery is a whole trajectory. For example, Gaffney et al. [24] propose a model-based clustering algorithm that groups trajectories with overall similar routes using a regression mixture model and the EM algorithm. This algorithm generate clusters of trajectories with respect to the overall distances among whole trajectories.

In contrast, *partial trajectory patterns* consider partial trajectories in the pattern discovery process. The idea behind this approach is that the clustering of whole trajectories may not detect trajectories with similar sub-trajectories. In general, a trajectory may be long and complex. Hence, even though some parts of trajectories are similar, the full trajectories might not be similar. Based in this view, Lee et al. [42] propose to partition trajectories into line segments and to build groups of close trajectory segments. Recent studies on trajectory clustering algorithms, such as moving clusters [36], flocks [11], convoys [35], and swarms [44], also consider patterns that appear in sub-trajectories.

Individual Vs. Group Patterns

Another distinction relates to whether a set of individual trajectories are retrieved that satisfy a pattern specified in query, called *individual trajectory pattern* retrieval [28, 32, 48], or whether sets of trajectories are retrieved so that the trajectories in a set exhibit a similar pattern according to some specific notion of pattern, called *group trajectory pattern* discovery [27, 33]

So-called *spatiotemporal pattern queries* [28, 48] illustrate well the retrieval of trajectories that satisfy a particular pattern as specified in a query. Examples of such queries include the finding of flights that descended into an airport, but did not land; identifying flights that had to make several approaches before entering the airport; and discovering vessels that changed course to avoid a collision.

The discovery of groups of trajectories that share patterns [57] is fundamentally different in that the objects returned are to be similar to each other according to a given notion of similarity rather than similar to a query pattern. The discovery of group patterns may enable different forms of sharing among the objects who have similar trajectories, e.g., in car pooling or delivery truck logistics. This class of patterns include concurrence [39], trend-setter [39], flock [11, 27, 26], leadership [5, 4], convergence [11], encounter [11], convoy [33], and swarm [44]. We describe each of these pattern concepts in detail in Sections 5.4 and 5.5.

5.2.2.4 Constrained Trajectory Patterns

Some trajectory patterns can be associated with constraints along the spatial and temporal dimensions.

Spatial Constraints: Movement on Spatial Networks

Many types of objects move in a spatial network, such as a road network, a rail network, or the kind of network made up by the corridors used by commercial aircraft. Since those objects are always located somewhere in the networks, raw trajectory data is typically modeled as or transformed to network-based trajectories, e.g., edge sequences in a road network graph [34, 60, 18]. As a result, trajectory patterns of network-constrained objects also have different forms and sometimes carry different semantics from the pattern types generally considered in unconstrained object trajectories.

As an example, Figure 5.4(a) illustrates a road network, with its graph model shown in Figure 5.4(b). All junctions form vertices, and each edge contains the corresponding road segment's information such as a weight w (distance). In this

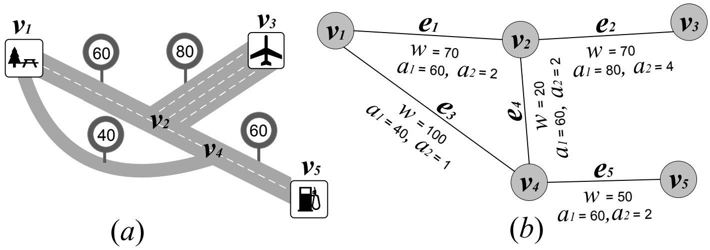


Fig. 5.4 An example of road network-constrained trajectory model.

network model, an object's location is represented as a tuple $l = (e,d,t)$, meaning that the object is located on the edge $e = (v_i, v_j)$, at distance d from v_i at time t. Table 5.1 shows an example of network-constrained objects' trajectories over 2 days.

object	trajectory ID	sequence of network positions
o_1	1	$(e_1,1.2,t_1),(e_1,2.9,t_2),(e_2,0.2,t_3)$
o_1	2	$(e_1,1.3,t_2),(e_1,3.4,t_3)$
o_2	3	$(e_3,7.6,t_1),(e_5,1.5,t_2),(e_5,5.3,t_3)$
o_2	4	$(e_1,1.2,t_1),(e_1,2.9,t_2),(e_2,0.2,t_3)$

Table 5.1 An example of network-constrained trajectories.

Given such network constrained trajectories, trajectory patterns reveal hidden, but useful, information. Finding popular routes, for example, can indicate reliable paths when drivers travel to unfamiliar destinations, as well as suggest higher-quality roads for truck delivery services [18]. They also enable location prediction that in turn can enable services that report relevant traffic conditions and upcoming points of interest (POIs) such as gas stations to users [12, 34]. In addition, Internet map services (e.g., Google Maps, Yahoo Maps) can be enriched based on trajectory patterns.

Temporal Constraints: Periodicity

In the real world, various types of objects exhibit periodic movement patterns. For example, many individuals go to work every weekday following the same or similar routes each day, public transportation is governed by time schedules, and animals annually migrate to reproduce or seek warmer climates. Findings in the literature suggest that car drivers tend to follow regular trajectories more than 70% of the time [37].

Periodic pattern mining of trajectory data concerns the discovery of periodic object behavior [47, 45], i.e., objects that follow the same routes (approximately) over regular time intervals. These periodic patterns provide an insight into, and concise explanation of, periodic behaviors (e.g., daily, weekly, monthly, and yearly) across long movement histories.

Periodic patterns are also useful for compressing movement data [14], since they summarize movement trajectories into a compact format. In addition, periodic patterns can serve as a basis for future-movement prediction [32]. Moreover, if an object fails to follow an established, regular periodic behavior, this could be a signal of an abnormal environmental change or an accident.

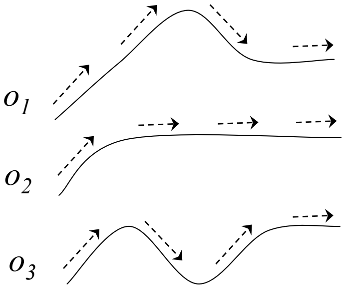
When considering object movement, it is typically unreasonable to expect an object to repeat its behavior exactly during each time period considered. This implies that patterns to identify should not be rigid, but that object behavior should be allowed to differ slightly from one period to the next while still resulting in a pattern. Next, behaviors that make up patterns may also be shifted in time (e.g., due to traffic delays).

The approximate nature of patterns in the spatiotemporal domain increases the complexity of mining tasks. In addition, the periods that yield patterns may be unknown. Further, multiple periods (e.g., *day* and *week*) may exist that yield different patterns in the same data. As a result, periodic pattern mining of trajectory data takes into account a wide variety of modeling approaches as well as efficient discovery algorithms.

5.3 Relative Motion Patterns

Given a collection of trajectories, it is challenging to capture and compare motion events of individuals and groups of individuals. To facilitate trajectory data analysis, the analysis concept RElative MOtion (REMO) by Laube *et al.* [39, 40, 38, 41] enables the identification of similar movements in a collection of moving object trajectories. The phrase "relative motion" refers to the relationships among motion attributes of different moving objects over space and over time.

REMO aims to enable the formulation and discovery of meaningful trajectory patterns based on *motion attributes* (i.e., speed, change of speed or motion azimuth), as extracted from raw trajectory data. Figure 5.5 demonstrates the process of building motion azimuth-based trajectories from raw trajectories. The left part of the figure shows trajectories of three objects moving in two-dimensional space. We then consider eight possible movement directions (motion azimuth): $\uparrow, \nearrow, \rightarrow, \searrow, \downarrow, \swarrow, \leftarrow, \nwarrow$. The right part shows the motion azimuth-based representation of the trajectories, where a trajectory is represented by a direction for each of four time points. This latter space of trajectories is called the analysis space in REMO.



(a) raw trajectories

	t_1	t_2	t_3	t_4
o_1	$\nearrow$	$\nearrow$	$\searrow$	$\rightarrow$
o_2	$\nearrow$	$\rightarrow$	$\rightarrow$	$\rightarrow$
o_3	$\nearrow$	$\searrow$	$\nearrow$	$\rightarrow$

(b) motion-azimuth trajectories

Fig. 5.5 Motion azimuth-based trajectory representation.

REMO defines three sets of relative motion patterns: *basic motion patterns, spatial motion patterns*, and *aggregate/segregate motion patterns*. These pattern concepts serve as fundamental background for subsequent studies of moving objects patterns. This section summarizes the key ideas in REMO.

5.3.1 Basic Motion Patterns

The first set of REMO patterns describe motion events and patterns that explicitly disregard the absolute position of the moving objects. This means that a set of geographically distant objects can be identified as belonging to the same group, as long as they exhibit the same predefined motion properties. The basic motion patterns include (i) patterns over time, (ii) patterns across objects, and (iii) patterns over time and across objects. Such patters are exemplified next.

Constance
This pattern is defined as a sequence of equal motion attribute values for some consecutive time points. As an example in Figure 5.5, object o_2 moves with a constant motion azimuth $\rightarrow$ during the interval of $[t_2, t_4]$.

Concurrence
A concurrence motion pattern captures the incidence of having multiple objects with similar motion attributes. For example, in Figure 5.5, objects o_1, o_2 and o_3 move with the same motion azimuth $\nearrow$ at time t_1.

Trendsetter
This motion pattern attempts to find objects that anticipate a certain motion pattern that is shared by a set of other objects in future. Thus, this complex pattern combines

the simple patterns concurrence and constance. For example, o_2 anticipates at t_2 the motion azimuth $\rightarrow$ that is shared by all other objects at time t_4.

5.3.2 Spatial Motion Patterns

Based on the motion patterns in the set of patterns just covered, REMO identifies three important trajectory patterns that are all classified as *spatial motion patterns*. This second set of motion patterns includes spatial constraints regarding the absolute positions of the moving objects [40, 5], meaning that certain objects' motions appear within a spatial range (e.g., rectangle, circle, or ellipse) for some duration of time. Therefore, these patterns capture objects that are geographically close to one another, which is different from the basic motion patterns.

Figure 5.6 illustrates examples of the following three spatial motion patterns. Each spatial constraint is a circle.

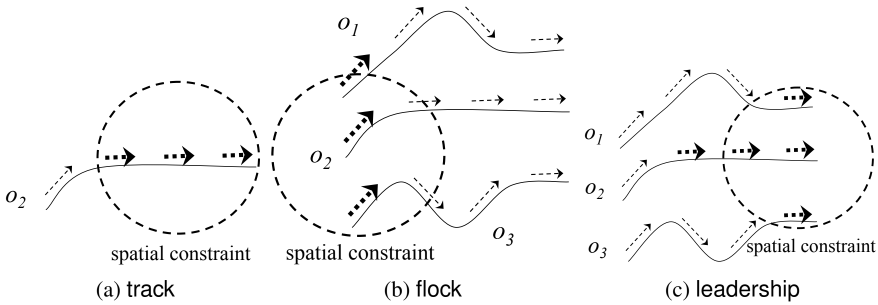


Fig. 5.6 Examples of REMO spatial motion patterns.

Track

Given a spatial range, e.g., a disc, a track motion pattern finds objects who travel within the range while keeping the same motion. For instance, an airplane that flies at constant speed and direction can be captured by the track motion pattern. Essentially, track combines constance pattern and a spatial constraint.

Flock

The concept of flock pattern in REMO covers not only a group of animals that live, travel, or feed together (e.g., a pride of lions, a school of fish, a gam of whales, a gaggle of geese, a murder of crows, or a swarm of insects), but also a group or crowd of people that move together.

In order to capture this concept using the basic motion patterns, the definition of flock consists of the concurrence pattern combined with a spatial constraint.

Intuitively, this patter may help identify groups of objects that travel together, i.e., within spatial proximity, for some duration of time.

Leadership
Leadership is a widespread phenomenon in social settings. In particular, the animal behavior research community studies the general topic of group decision-making in animals, searching for evidence for groups led in their activities by some dominant individuals, e.g., dominant breeding wolves frequently show significant frontal leadership, leading the pack during travel.

The leadership pattern corresponds to the trendsetter pattern, but includes a spatial constraint. For example, followers must lie within a given geographical region when they join the motion of the trendsetter. As a result, this pattern captures well the case where a group of people are under the leadership of one person.

5.3.3 Aggregate/Segregate Motion Patterns

The third set of relative motion patterns describe aggregation and segregation of objects' movements.

Convergence
This pattern describes a set of m objects during a time interval k that share motion azimuth vectors intersecting within a given spatial range, e.g., a disc with radius r. This pattern captures the behavior of a group of objects that converge in a certain region. An example of this pattern is wild animals that are heading in a synchronized fashion for a mating place.

Encounter
Suppose that some antelopes distributed over a field are heading for a location. At some later time, they will thus meet at the location, according to their current motions. To capture such an extrapolated (future) meeting within a spatial range, the encounter pattern is defined as a set of m objects that will arrive in a given spatial range r concurrently k time points later (i.e., the extrapolations of the objects' current motions intersect with r).

Divergence
This pattern is an opposite concept of convergence, integrating a spatial divergence pattern with the temporal constraint of a preceding meeting in a region. The graphical representation of the divergence pattern is that of "heading backwards" instead of forwards, relative to the direction of motion.

Breakup
Like the convergence pattern is an opposite concept of convergence, the concept of breakup is an opposite of the concept of encounter. This pattern captures objects' behaviors, such as departing from a meeting point.

5.3.4 Discussion

Although the REMO patterns constitute important mining concepts for trajectory data, they pose several open problems that should be addressed. We cover such open problems briefly.

- In the REMO framework, it is difficult to determine an absolute distance between two objects because the pattern discovery process is performed over the motion attributes (i.e. speed, change of speed, or motion azimuth) that are derived from raw trajectories. As a result, some pattern analysis tasks, such as finding the *k* nearest neighbor objects of a given object (trajectory), are difficult to support in REMO.
- Although the motion attributes in REMO encompass the objects' speeds and changes of speeds, the objects' motions are mainly analyzed considering the motion azimuths that capture the direction of a trajectory at a point in time. Full motion analysis considering all the motion attributes simultaneously is indeed not a simple task, but requires complex mechanisms and heavy computation.
- The default motion azimuths in REMO consists of eight different angles. Using a finer or coarser angle granularity would substantially impact the effectiveness and efficiency of trajectory pattern discovery. For example, the classification of motion azimuths into only the two classes East and West would reveal a lot of presumably meaningless constancy patterns. In contrast, every constancy pattern found with 360 azimuth classes may be unnecessary for typical applications. It is non-trivial to determine an appropriate angle granularity for a given trajectory data set.
- During the detection of relative motion patterns, time intervals with uncertain or missing data points may reduce the accuracy and effectiveness of pattern discovery significantly. In addition, the REMO framework assumes all trajectories have the same sampling rate as the granularity used in the analysis task, which may not hold for real-world trajectory data. Before employing the concept of relative motion patterns, pre-processing steps are required (e.g., interpolating missing samples and smoothing trajectories) in order to ensure a uniform time granularity.

5.4 Disc-Based Trajectory Patterns

Subsequent to the introduction of relative motion patterns by Laube *et al.* , a substantial body of studies have continued the study of trajectory data pattern mining [11, 35, 27, 54, 44, 8, 5, 4]. These subsequent studies have redefined, extended, and further developed the concepts of trajectory patterns substantially. The advances over the original relative motion pattern concepts relate mainly to three aspects.

1. *Distance-based trajectory analysis*: the core idea of REMO—i.e., extracting and utilizing motion features of trajectories for pattern analysis—is no longer

considered in further studies. Instead of using motion attributes, the subsequent studies consider Euclidean distance for measuring the proximity of positions of trajectories at time points. As a result, the importance of the basic relative motion patterns constance, concurrence, trend-setter decreases in trajectory analysis.
2. *Disc-based spatial range*: while REMO allows regions with arbitrary geometric shapes (e.g., ellipse, rectangle, disc) to be used as spatial ranges in motion patterns (Section 5.3.2), the subsequent studies of REMO take into account only circular ranges, i.e., discs. This approach is easy to apply, and discs capture intuitively the ranges of moving-object groups.
3. The REMO framework lacks time constraints, meaning that the relative motion patterns do not include an explicit parameter for specifying the time lengths of motion patterns. A relative motion pattern that, e.g., lasts for a few seconds may be too short to identify a common behavior among objects. Yet such a pattern can be identified with REMO. Subsequent studies include an explicit time duration constraint in order to alleviate this problem.

Built on the above principles, a rich body of studies have refined or redefined the original concepts and definitions of relative motion patterns. They have also introduced a set of new pattern types, mainly based on the REMO concept.

We categorize these newly introduced patterns into two classes:

- *Prospective patterns* that are likely to occur some time in the near future. This class covers encounter and convergence.
- *Flock-based* patterns that correspond to the original flock pattern of REMO. This class includes redefinitions of flock, meet, and leadership.

This section discusses in detail concepts that relate to these patterns.

5.4.1 Prospective Patterns

Gudmundsson *et al.* [27] revisit the encounter and convergence patterns in REMO, providing generic definitions based on the geometric arrangements of the moving objects. The resulting new patterns thus do not require the motion attributes in the REMO framework. The new pattern definitions consider the future trajectories of moving objects, meaning that the new encounter and convergence patterns capture future events that are likely to occur, based on the current motions of moving objects. Specifically, they are defined as follows.

Encounter (m,r)
Satisfied by a group of at least m objects that will arrive simultaneously in a disc with radius r, assuming that they keep their current speeds and directions.

Convergence (m,r)
Satisfied by a group of at least m objects that will pass through a disc with radius r

(not necessarily at the same time), assuming that they keep their current movement directions.

Figure 5.7 demonstrates the current and future trajectories obtained from five objects o_1, o_2, o_3, o_4, and o_5. The arrows show the current positions and their current motions, i.e., speed and direction; the dotted lines indicate the objects' future trajectories based on the current motions. According to the future trajectories, o_2, o_3, o_4, and o_5 will pass the disc r, forming a convergence pattern if given an m that does not exceed 45. If these objects arrive at r at the same time, they also satisfy the encounter pattern with the same parameters.

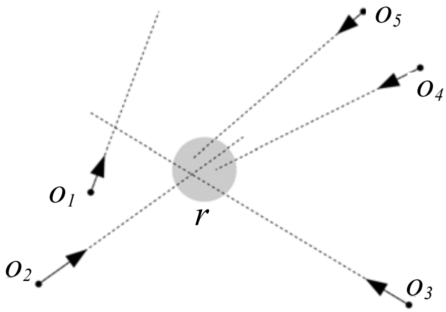


Fig. 5.7 Examples of the encounter and convergence patterns.

5.4.2 Flock-Driven Patterns

Gudmundsson *et al.* [11, 27, 26, 3, 54] have studied extensions of the flock pattern introduced by Laube *et al.* [39, 40] intensively. The studies have resulted in a wide range of interesting trajectory pattern concepts and discovery techniques. This section summarizes key ideas in relation to the extensions and variants of flock.

Flock (m, r, k)
In one study [11], two types of flock patterns are defined: one concerns continuous object movement, while the other concerns discrete object movement. The latter is widely used in database and data mining research. A discrete flock occurs if at least m objects move together for at least k consecutive time points while staying within a disc with radius r.

Based on this flock concept, two interesting variants, meet [11] and leadership [5], are defined:

Meet (m, r, k)
A meet pattern occurs if at least m objects stay together in a *stationary* disc with

radius r for at least k consecutive time points. Unlike the definition of flock, the disc specified in meet has a fixed location. Thus, the concept of meet resembles a past variant of encounter.

Figure 5.8 illustrates the concepts of flock and meet. In Figure 5.8(a), a flock of objects o_1, o_2, and o_3 is found during the time points [t_7, t_8, and t_9, while the meet pattern of objects o_1 and o_2 is identified during the time points $t_3, \ldots, t_8$ (Figure 5.8(b)).

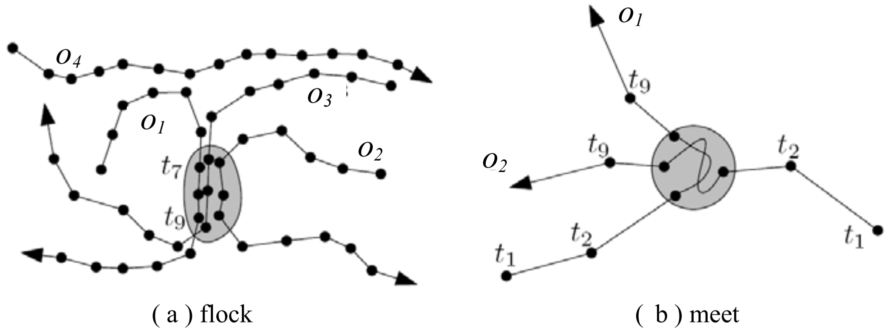


Fig. 5.8 Examples of the flock and meet patterns.

Leadership (m, r, k)

Andersson *et al.* [4] present a clear concept and definition of a leadership pattern. Their leadership pattern is essentially an extension of flock. Yet it captures the spatial constraint of the leader being ahead of the followers. Formally, the pattern occurs when a set of at least m objects move together for at least k consecutive time points while staying within a disc with radius r, and when at least one of the objects is/was heading in the leader's direction.

Andersson *et al.* [5] also introduce a variant of leadership where two new parameters are added to the definition of leadership. In Leadership (m, r, k, α, β), parameter α influences the spatial extent of a pattern, and parameter β determines the spatial characteristics of a pattern.

5.4.3 Discussion

The disc-based trajectory patterns are satisfied by groups of objects that move together within a disc with some user-specified extent. As a result, the chosen disc extent has a substantial effect on the result of the discovery process. Although the disc concept is intuitive, it raises potential problems when specifying trajectory patterns, as discussed next.

- The selection of a proper disc size turns out to be difficult, as situations can occur where objects that intuitively belong together or do not belong together are not quite within any disk of the given size or are within such a disk. In Figure 5.9, for example, all objects travel together in a natural group. However, object o_4 does not enter the disc and is not discovered as a member of the flock. This problem occurs because what constitutes a flock is very sensitive to the user-specified disc size, which does not take the data distribution into account. Indeed, if a flock must contain at least 4 objects, no flock would be found in this example.
- For some data sets, no single appropriate disc size may exist that works well for all parts of the $(\text{space}, \text{time})$ domain. A herd of animals, for instance, consists of individual that move together, but the herd may expand or contract over time. This can not be captured by a fixed-size disc.
- The use of a circular shape may not always be appropriate. For example, suppose that two different groups of cars move across a river and each group has a long linear form along roads. A sufficient disc size for capturing one group may also capture the other group as one flock. Ideally, no particular shape should be fixed apriori.

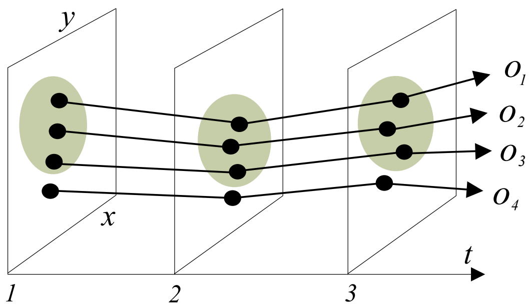


Fig. 5.9 Lossy-flock problem.

5.5 Density-Based Trajectory Patterns

In order to avoid the size and shape restrictions inherent to disc-based trajectory patterns, *density-based trajectory patterns* have been introduced. These employ density concepts [22] that allow the capture of generic trajectory pattern of any shape and any extent. In this section, we first review the concept of density-based clustering and then describe in detail the key concepts and definitions relating to density-based trajectory patterns.

5.5.1 Density Notions

As a precursor to defining density-based trajectory patterns, we need to understand the notion of density connection [22]. The main advantage of this concept is that it is able to capture clusters of any size and shape, as long as the cluster members meet certain distance-related conditions.

- Given a distance threshold e and a set of points S, the *e-neighborhood* of a point p is given as $NH_e(p) = \{q \in S \mid D(p,q) \leq e\}$.
- Given a distance threshold e and an integer m, a point p is *directly density-reachable* from a point q with respect to e and m if $p \in NH_e(q)$ and $|NH_e(q)| \geq m$.
- A point p is said to be *density-reachable* from a point q with respect to e and m if there exists a chain of points p_1, p_2, ..., p_n in set S such that $p_1 = q$, $p_n = p$, and p_{i+1} is directly density-reachable from p_i with respect to e and m, $i \in \{1,\ldots,n-1\}$.
- Given a set of points S, a point $p \in S$ is *density-connected* to a point $q \in S$ with respect to e and m if there exists a point $x \in S$ such that both p and q are density-reachable from x with respect to e and m.

Figure 5.10(a) exemplifies a density-connected cluster when assuming that $m = 4$. Each dashed circle indicates an e-neighborhood of some object. For instance, o_1 is directly density reachable from o_3 with respect to threshold e and $m = 4$ because it belongs to the e-neighborhood of o_3 that contains a total of 4 objects. Also, o_1 and o_9 are density-connected through the following chain of points: $\langle o_3, o_4, o_6, o_7 \rangle$. The collective extent of the cluster can exceed e, and the cluster can have arbitrary shape. Since the parameter m is used to determine whether a point is directly density reachable from another, the size of a density-connected cluster is at least m. In addition, the cluster is preserved at time $t = 2$ in Figure 5.10(b), although the size of the cluster and the topology of the cluster members are changed.

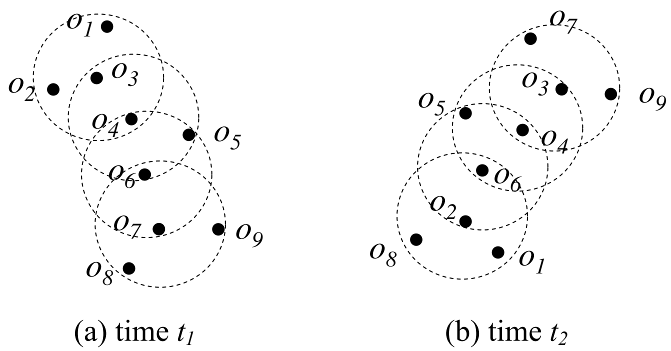


Fig. 5.10 An example of a density-connected cluster.

As shown in the above example, the definition of density-connection permits us to capture a group of "connected" points with arbitrary shape and extent. It thus allows us to overcome the main drawbacks of the concept of disc-based trajectory patterns. We proceed to introduce trajectory patterns built upon the concepts of density-connected objects.

5.5.2 Moving Objects Clustering

Recently, a wide variety of clustering concepts and algorithms for trajectory data have appeared in the literature. The different proposals generally assume different data and application requirements, and they have different advantages and disadvantages. In this section, we present a clear, systematic, and comprehensive overview of recent works on the clustering of trajectories in order to offer a good foundation for subsequent studies of trajectory pattern mining.

5.5.2.1 TRACLUS

Lee *et al.* introduce *trajectory clustering* [42] for the purpose of grouping trajectory segments. Specifically, they proposed a partition-and-group framework TRACLUS that clusters trajectories in the following three steps:

- *Partitioning*: each trajectory is partitioned into a set of line segments. This process finds the points where the behavior of a trajectory changes, called its characteristic points. The process adopts the minimum description length (MDL) principle, which is used widely in information theory.
- *Grouping*: trajectory segments that are close to each other according to a certain distance measure are grouped into a cluster. For this process, Lee *et al.* define a set of distance measures for trajectory segments (details are presented in Section 5.6.1) that are used for density-based clustering using the DBSCAN algorithm [22]. This allows the clusters of trajectories obtained by TRACLUS to form any shape and size.
- *Representing*: given a cluster , this process derives a representative trajectory for the cluster that describes the overall movement of the trajectory partitions that belong to the cluster. Basically, this process averages the lengths and angles of the entire trajectory segments belonging to the cluster, thus constructing a new trajectory segment, i.e., a representative trajectory.

Figure 5.11 illustrates the result of the above three processing steps of the partition-and-group framework for trajectory clustering. The trajectory segments shown in grey in Figure 5.11(b) denote those that are not included in the result clusters c_1 and c_2. The dotted line segments "in" each cluster in Figure 5.11(b) show the representative trajectory segments of c_1 and c_2.

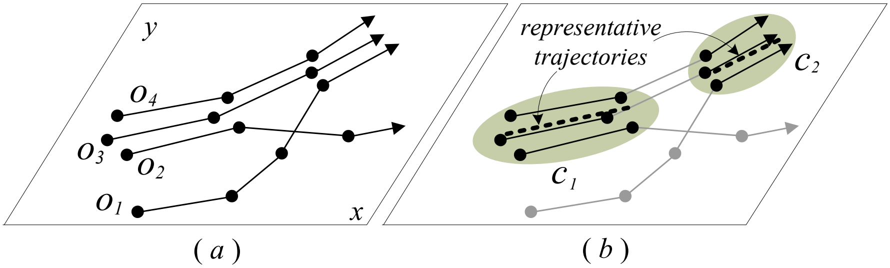


Fig. 5.11 An example of trajectory clustering: (a) raw trajectories; (b) clustered trajectory segments.

Notice that TRACLUS does not consider the temporal aspects of the trajectories while partitioning and grouping trajectories. As a result, some objects can belong to the same group even though they have never traveled close together (i.e., traveled at the same time). That is, TRACLUS performs spatial trajectory pattern clustering as opposed to spatio-temporal trajectory pattern clustering, as described in Section 5.2.2.2.

5.5.2.2 Moving Cluster

Kalnis *et al.* introduce the concept of moving cluster [36] that is defined as a set of objects that move close to each other for a time duration. It is a sequence of spatial clusters appearing during consecutive time points, such that the portion of common objects in any two consecutive clusters is not below a given threshold θ, i.e., $\frac{|c_t \cap c_{t+1}|}{|c_t \cup c_{t+1}|} \geq \theta$, where c_t denotes a cluster at time t. As an example, o_1, o_2, and o_3 in Figure 5.12 form a moving cluster if $\theta \geq \frac{3}{4}$ (i.e., requiring at least 75% overlapping objects between two consecutive clusters in time).

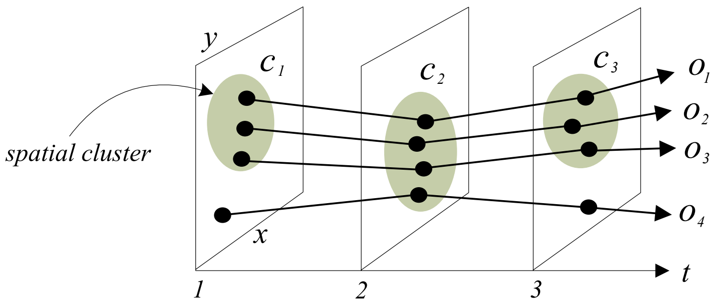


Fig. 5.12 An example of moving cluster.

It is worth noticing that the concept of moving cluster is not effective at capturing groups of objects that traveled together, since a moving cluster can be formed as long as two snapshot clusters have at least θ overlap, regardless of which objects traveled together from the beginning to the end of the trip.

5.5.2.3 Convoy

The concept of convoy [35, 33] employs the notion of density connection [22] in order to enable the formulation of arbitrary shapes of groups. More specifically, given a set of trajectories O, an integer m, a distance value e, and a lifetime k, a convoy is defined as a group that has at least m objects who are density-connected with respect to distance e and cardinality m during k consecutive time points. Each convoy is associated with a time interval during which the objects in the group traveled together.

Figure 5.13 shows polylines that represent the trajectories of three objects o_1, o_2, and o_3, during the time interval from t_1 to t_4. Consider the convoy specified by the parameters $m = 2$ and $k = 3$ issued over the trajectories in the figure. The result is $\langle o_2, o_3, [t_1, t_3] \rangle$, meaning that o_2 and o_3 form a convoy during the consecutive time points from t_1 to t_3.

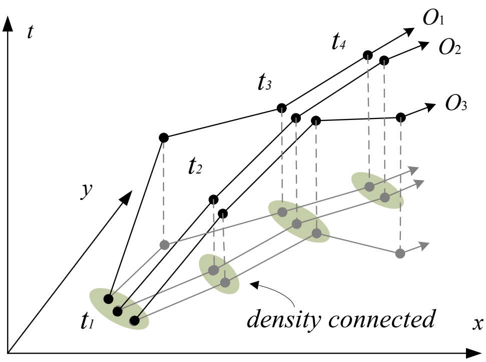


Fig. 5.13 An example of a convoy.

The conceptual difference between convoy and moving cluster is demonstrated in Figure 5.14(a). Here, objects o_2, o_3, and o_4 form a convoy with 3 objects during 3 consecutive time points. The overlap between the cluster at time 1, c_1, and the cluster at time 2, c_2, is $\frac{3}{4}$. Thus, if we set $\theta = 1$ (i.e., require 100% overlapping clusters), the above objects will not be discovered as a moving cluster. Next, in Figure 5.14(b), if we set $\theta = \frac{1}{2}$ then c_1, c_2, and c_3 become a moving cluster. However, they do not form a convoy.

convoy Variants

Since the concept of convoy was originally introduced [35], subsequent studies

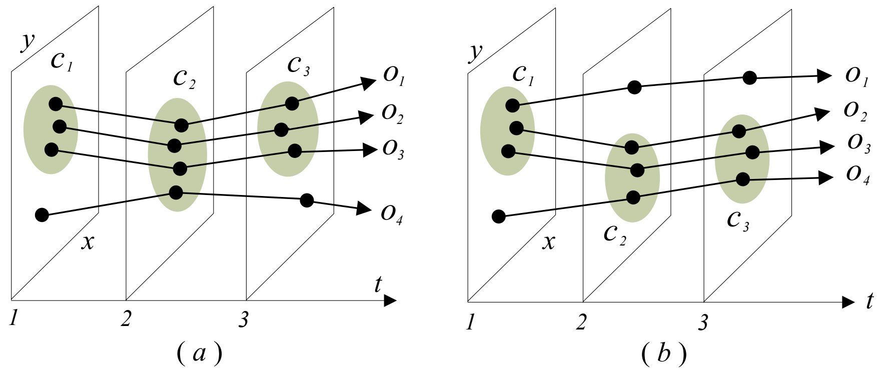


Fig. 5.14 Comparison between moving cluster and convoy.

have supplemented the original concept. Aung and Tan [8] introduce two variants of convoys: dynamic convoys allow their members to be absent briefly during the convoy lifetime, while evolving convoys are allowed to grow and shrink in cardinality during their lifetimes, which reduces the number of convoys that have large sets of overlapping objects and similar lifetimes.

In addition, Yoon and Shahabi [59] report that the original concept of convoy may miss some convoys that still satisfy with the definition, when two separate convoys are merged into one larger, union convoy. To capture this case, they introduce a variant of the convoy concept, termed a valid convoy.

5.5.2.4 Swarm

The convoy concept has a strict restriction regarding time: all convoy objects must be together for all of the consecutive time points in a lifetime of their convoy. For example, consider the group of moving objects o_1, o_2, o_3, and o_4 shown in Figure 5.15, where the first three objects belong to the same cluster for three time points considered and where object o_4 was apart from the others only during the second time point. Due to the strict time constraint, o_4 cannot belong to the same convoy as the first three objects although all the objects may be viewed as traveling together.

In order to overcome this drawback of convoy, Li *et al.* introduce a new trajectory pattern type, swarm [44], that extends the concept of convoy by relaxing the consecutive-time constraint. Specifically, the definition of swarm replaces the parameter k of convoy with k_{min}, such that k_{min} denotes a minimum of time duration to form a moving object cluster, regardless of the consecutiveness in time. This allows an individual moving object to temporarily leave its group as long as it is close to other group members for most of the time.

The concept of swarm is similar to that of dynamic convoy [8]; however, it differs in its discovery technique. swarm also has a definition similar to that of group pattern [57] in the sense of identifying a group of objects that travel together while

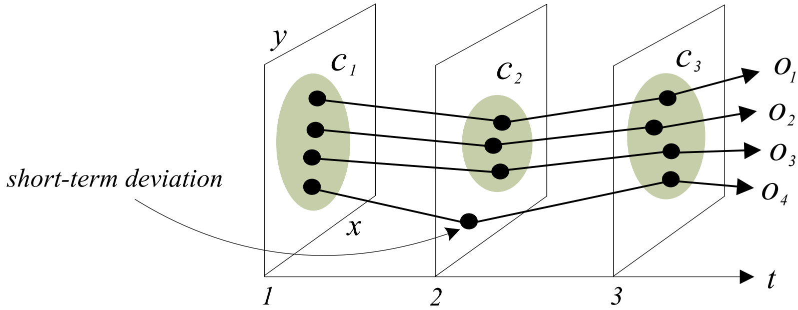


Fig. 5.15 An example of a swarm.

allowing relaxation of the time constraint. Nevertheless, group pattern belongs to the class of disc-based trajectory patterns.

Variants of Swarm

The discovery of swarms in a large collection of trajectories may result in a number of similar or redundant clusters in time or objects. To avoid finding such redundant swarms, the concept of closed swarm is introduced [44].

A follower trajectory pattern [46] identifies objects that follow a swarm with a certain time lag, which may be useful to capture a case such as when a animal follows a herd at a certain distance. This is similar to the concept of leadership in the sense that some objects spatially intersect a certain cluster with some time delay.

5.5.3 Discussion

Table 5.2 summarizes the key features of each concept of moving object clustering.

	Pattern space	Proximity-based	Online adaptation	Variants
TRACLUS	spatial	trajectory segments	difficult	TraClass [43]
Moving cluster	spatiotemporal	snapshot points	easy	
Convoy (CMC)	spatiotemporal	snapshot points	easy	dynamic/evolving
(CuTS)	spatiotemporal	trajectory segments	difficult	/valid convoy
Swarm	spatiotemporal	snapshot points	difficult	closed swarm, follower

Table 5.2 Summary of moving object clustering algorithms.

5.6 Methods for Mining Trajectory Patterns

We review distance functions for measuring the distance between two trajectories and then examine techniques for the mining of patterns from trajectories.

5.6.1 Trajectory Distance Measures

We use the notation $P_{1..n}$ for a trajectory that consists of a sequence of time-referenced point locations $P_1, P_2, \cdots, P_n$, where P_t is the point at time t (in the interval $[1,n]$). A trajectory pattern mining task (e.g., trajectory clustering) often needs to compute the "closeness" of two trajectories, which is captured by a distance function. We classify existing distance functions as global distance measures or local distance measures. In the sequel, we consider measures in each class in turn and discuss how the individual measures capture the characteristics of trajectories.

5.6.1.1 Global Distance Measures

A global distance function defines the overall distance between two trajectories with respect to all points in those trajectories.

Euclidean Distance

A simple approach to measuring the distance between two trajectories is to compute the sum of the Euclidean distances between all corresponding pairs of point locations in the two. This definition assumes that a pair of argument trajectories P and Q have the same length—i.e., they are sampled at the same times.

Specifically, the Euclidean distance between such trajectories $P_{1..n}$ and $Q_{1..n}$, $D_{Euclid}(P_{1..n}, Q_{1..n})$, is defined as the sum of their point distances at each sampling time t, i.e.,

$$D_{Euclid}(P_{1..n}, Q_{1..n}) = \sum_{t=1}^{n} ||P_t - Q_t|| \,,$$

where the Euclidean distance between two points is defined as:

$$||P_t - Q_t|| = \sqrt{(P_t.x - Q_t.x)^2 + (P_t.y - Q_t.y)^2}$$

In cases where only relative distances, not absolute distance, are important, the squared Euclidean distance may be used as an alternative to the Euclidean distance. The squared Euclidean distance preserves the ordering of the Euclidean distance and is attractive because it is easier to compute.

Alignment-based Distance

The Euclidean distance measure is affected by noise and distortion that may obscure a trajectory. Thus, it may be argued that the Euclidean distance is unable to capture the inherent distance between trajectories. As a result, various alignment-based distance measures have been proposed for computing the distance between a pair of trajectories. These measures also lift the rigid length assumption inherent to the Euclidean distance and permit the comparison of trajectories with different lengths.

Dynamic Time Warping (DTW) [58] is defined by the following recurrence equation. It attempts to align two trajectories in such a way that their overall distance is minimized. The computation of DTW requires dynamic programming and it takes $O(n \cdot m)$ time.

$$DTW(P_{1..n}, Q_{1..m}) = ||P_n - Q_m|| + \min \begin{cases} DTW(P_{1..n-1}, Q_{1..m-1}) \\ DTW(P_{1..n-1}, Q_{1..m}) \\ DTW(P_{1..n}, Q_{1..m-1}) \end{cases}$$

In the equation, $P_{1..n-1}$ denotes the sub-trajectory of $P_{1..n}$ that covers the time points from 1 to $n-1$ only.

The dynamic time warping distance is not a metric, as it does not satisfy the triangle inequality. In contrast, a recent, related measure called the Edit Distance on Real sequence (EDR) [16] is a metric. It satisfies the triangle inequality, so it can be exploited for pruning unnecessary trajectories effectively during query processing.

When compared to the above distance measures, the Longest Common SubSequence measure (LCSS) [55] is more robust to noise. However, it requires the user to specify two parameters δ and ε as tolerances for two points to match with respect to time and space. This measure is defined by the recurrence equation next.

$$LCSS(P_{1..n}, Q_{1..m}) = \begin{cases} 0 & \text{if } n = 0 \vee m = 0 \\ 1 + LCSS(P_{1..n-1}, Q_{1..m-1}) & \text{if } |n-m| \leq \delta \\ & \wedge ||P_n - Q_m|| \leq \varepsilon \\ \max\{LCSS(P_{1..n-1}, Q_{1..m}), \} \\ \qquad LCSS(P_{1..n}, Q_{1..m-1}) & \text{otherwise} \end{cases}$$

A recent distance measure called the Edit distance with Real Penalty (ERP) [15] is also designed to handle noise and it is even better than LCSS.

5.6.1.2 Local Distance Measures

Global distance measures capture the overall similarity between a pair of trajectories, not their local similarity during some short time interval. We proceed to consider local distance functions that capture the similarity between sub-trajectories.

MBR-Based Distance

A commonly used trajectory distance measure is derived based on the use of Minimum Bounding Rectangles (MBRs), which are often used to approximate trajectory segments and provide fast computation of trajectory distances. Let B_1 and B_2 be the MBR of the line segment L_1 and L_2, respectively. The distance $D_{min}(B_1, B_2)$ represents the minimum distance between any pair of points in B_1 and B_2. It is defined as:

$$D_{min}(B_1,B_2) = \sqrt{(\Delta(B_1.[x_l,x_u],B_2.[x_l,x_u]))^2 + (\Delta(B_1.[y_l,y_u],B_2.[y_l,y_u]))^2} \ ,$$

where the distance between two intervals is defined as:

$$\Delta([l_1,u_1],[l_2,u_2]) = \begin{cases} 0 & [l_1,u_1] \cap [l_2,u_2] \neq \emptyset \\ l_2 - u_1 & \text{if } u_1 < l_2 \\ l_1 - u_2 & \text{if } u_2 < l_1 \end{cases}$$

Trajectory-Hausdorff Distance

Lee *et al.* [42] propose a distance function that is a weighted sum of three terms:

$$D_{Hausdorff} = w_\perp \cdot d_\perp + w_\| \cdot d_\| + w_\theta \cdot d_\theta \ ,$$

where $w_\perp$, $w_\|$, and w_θ are the weights of the following components. They suggest that different applications may require different weights.

- The aggregate perpendicular distance ($d_\perp$) that measures the separation between two trajectories. where $d_{\perp,a}$ and $d_{\perp,b}$ are two perpendicular distances between
- The aggregate parallel distance ($d_\|$) that captures the difference in length between two trajectories.
- The angular distance (d_θ) that reflects the orientation difference between two trajectories.

Figure 5.16 illustrates these distance components between two line segments L_1 and L_2. Specifically, we have:

$$d_\perp = \frac{d_{\perp,a}^2 + d_{\perp,b}^2}{d_{\perp,a} + d_{\perp,b}}$$

$$d_\| = \min\{d_{\|,a}, d_{\|,b}\}$$

$$d_\theta = ||L_2|| \times \sin\theta$$

where $d_{\perp,a}, d_{\perp,b}$ are two perpendicular distances between L_1 and L_2, $d_{\|,a}, d_{\|,b}$ are two parallel distances between L_1 and L_2, and θ is the angle between L_1 and L_2.

Trajectory-Segment Distance

Jeung *et al.* [35] use two simple measures as the distance between two trajectory segments l'_1 and l'_2. These measures are designed to support efficient trajectory clustering.

The line distance $D_{LL}(\cdot,\cdot)$ is simply the shortest distance Euclidean distance between a pair of points located on the two argument line segments, and it is thus defined as follows.

$$D_{LL}(l'_1,l'_2) = \min_{p_i \in l'_1, b_j \in l'_2} ||p_i - b_j||$$

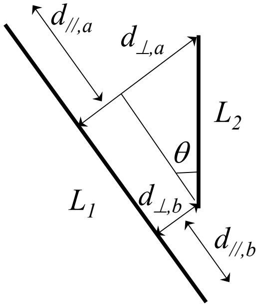


Fig. 5.16 Trajectory-Hausdorff distance.

Figure 5.17(a) shows two line segments l'_1 and l'_2. Here, l'_1 has the endpoints p'_1 and p'_4, corresponding to its locations at times t_1 and t_4. Similarly, l'_2 has endpoints b'_3 and b'_5, for its locations at times t_3 and t_5. The shortest distance between l'_1 and l'_2 is given by $D_{LL}(l'_1, l'_2)$. Note that this distance is purely spatial and ignores the temporal dimension.

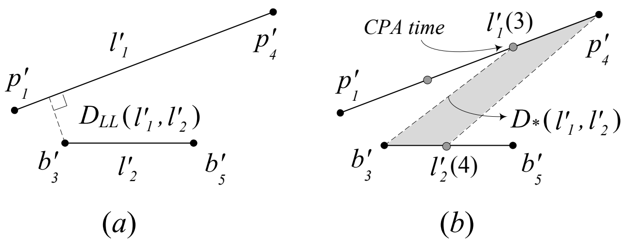


Fig. 5.17 Trajectory-segment distance examples.

In contrast, the tightened distance $D_*(\cdot,\cdot)$ is spatio-temporal and thus aligns two line segments based on time. It then measures the shortest distance between two trajectories as of the same time point and is defined as follows.

$$D_*(l'_1, l'_2) = \min_{p_i \in l'_1, b_j \in l'_2, i=j} ||p_i - b_j||$$

In the example in Figure 5.17(b), the tightened distance $D_*(l'_1, l'_2)$ is the distance between the locations $l'_1(3)$ and $l'_2(3)$, which are the derived locations of the two trajectories at time 3.

It is possible to compute the tightened distance $D_*(l'_1, l'_2)$ efficiently. Let $l'_p = \{p_u, p_v\}$ be a line segment having a time interval $l'_p.\tau = [u, v]$. The location of a point l'_p in the segment as of time $t \in [u, v]$ is defined as:

$$l'_p(t) = p_u + \frac{t-u}{v-u}(p_v - p_u)$$

Note that the terms $l'_p(t)$, p_u, and $(p_v - p_u)$ are 2D vectors representing locations.

We then introduce the *Closest Point of Approach* time, called the CPA time (t_{CPA}) [6]. This is the time when the distance between two dynamic objects is the shortest, considering their velocities. Let $l'_q = \{q_w, q_z\}$ be another line segment during $l'_q.\tau = [w,z]$. The CPA time for l'_p and l'_q is computed by:

$$t_{CPA} = \frac{-(p_u - q_w) \cdot (l'_p(t) - l'_q(t))}{|l'_p(t) - l'_q(t)|^2} \ ,$$

where, $l'_q(t)$, q_w, and $(q_w - q_z)$ are also location vectors.

Considering again the example in Figure 5.17(b), the gray region indicates that the common time interval of l'_1 and l'_2 is $[t_3, t_4]$. The *tightened* shortest distance $D_*(l'_1, l'_2)$ between the two segments is then taken as:

$$D_*(l'_1, l'_2)) = \begin{cases} D(l'_1(t_{CPA}), l'_2(t_{CPA})) & \text{if } t_{CPA} \in (l'_1.\tau \cap l'_2.\tau) \\ \infty & \text{if } l'_1.\tau \cap l'_2.\tau = \emptyset \end{cases}$$

hen the time intervals of the two argument segments do not intersect, i.e., $l'_1.\tau \cap l'_2.\tau = \emptyset$, their distance is ∞ .

The above equation can be expressed as a quadratic equation of t, and its minimum value can be found in constant time, regardless of the length of the common time interval $l'_1.\tau \cap l'_2.\tau$.

Observe that $D_*(l'_1, l'_2)$ is longer than $D_{LL}(l'_1, l'_2)$; hence, the line segments in Figure 5.17(b) have a lower probability of forming a cluster than do those in Figure 5.17(a). The tightened distance bounds thus improve the effectiveness of a filtering step that is applied as part of trajectory clustering.

5.6.2 Techniques for Efficient Pattern Discovery

It is difficult to discover trajectory patterns from a large trajectory database in an efficient manner. This process generally calls for the computation of sets of objects, which is more expensive than, e.g., spatio-temporal joins [10] that compute pairs of objects. In this section, we present various techniques from the literature that enable efficient discovery of trajectory patterns.

5.6.2.1 Raw Data Transformation

The REMO Matrix

The REMO framework enables the comparisons of the motion attributes of point objects over space and time, and it thus makes it possible to relate one object's motion to the motions of all other objects. More specifically, the given raw trajectory data is first transformed into a REMO matrix featuring motion attributes (i.e. speed, change of speed, or motion azimuth). The REMO framework then uses the REMO matrix to identify notable individual motion behaviors as well as events of distinct group motion behavior, e.g., patterns like constancy, concurrence, and trend-setter as discussed in Section 5.3.

Table 5.3 illustrates an example of a REMO matrix that supports efficient discovery of patterns as illustrated by the following examples.

- The constancy pattern appears as a consecutive row of identical values, e.g., object o_1 from time t_1 to t_4.
- The concurrence pattern appears as a column (not necessarily consecutive) of identical values, e.g., objects o_2, o_3, o_4 at time t_2.
- The trend-setter pattern is a constancy pattern followed by a concurrence pattern, e.g., object o_4 from time t_4 to t_6, then objects o_4, o_5, o_6 at time t_6.

	t_1	t_2	t_3	t_4	t_5	t_6
o_1	↑	↑	↑	↑	→	↗
o_2	↗	↓	↑	↖	←	↘
o_3	→	↓	←	↘	↗	
o_4	↘	↓	→	↙	↙	↙
o_5	←	↗	↖	↗	↑	↙
o_6	↖	↘	↓	↗	↖	↙

Table 5.3 An example REMO matrix.

In this approach, raw data is approximated and transformed into an analysis format (i.e., the motion matrix) optimized for pattern discovery. This accelerates the pattern discovery process substantially when compared with directly accessing raw trajectory data to retrieve relative motion patterns.

Trajectory Simplification

The work on convoy discovery [35, 33] applies the filter-and-refinement paradigm in order to reduce the overall computational cost. In the filtering step, the original trajectories are simplified and a clustering algorithm is applied on the simplified trajectories in order to obtain convoy candidates. The goal is to retrieve efficiently a superset of the actual convoys. In the refinement step, each candidate convoy is considered in turn, and clustering is performed on the original trajectories of the objects involved so as to determine whether they indeed form an actual convoy.

Specifically, given a trajectory represented as a polyline $o = \langle p_1, p_2, \cdots, p_T \rangle$, and a simplification tolerance δ, the Douglas-Peucker algorithm (DP) [20] is applied to derive a *simplified trajectory* o' such that o' has fewer points and o' deviates from o by at most δ at any point. Initially, DP composes the line $\overline{p_1 p_T}$ and finds the point $p_i \in o$ farthest from the line. If the distance $D_{PL}(p_i, \overline{p_1 p_T}) \leq \delta$, then the segment $\overline{p_1 p_T}$ is reported as the simplified trajectory o'. Otherwise, DP recursively examines the sub-trajectories $\langle p_1, \cdots, p_i \rangle$ and $\langle p_i, \cdots, p_T \rangle$, reporting the concatenation of their simplified trajectories as the simplified trajectory o'. Figure 5.18(a) illustrates three original trajectories. Their simplified trajectories (by using DP) are shown in Figure 5.18(b).

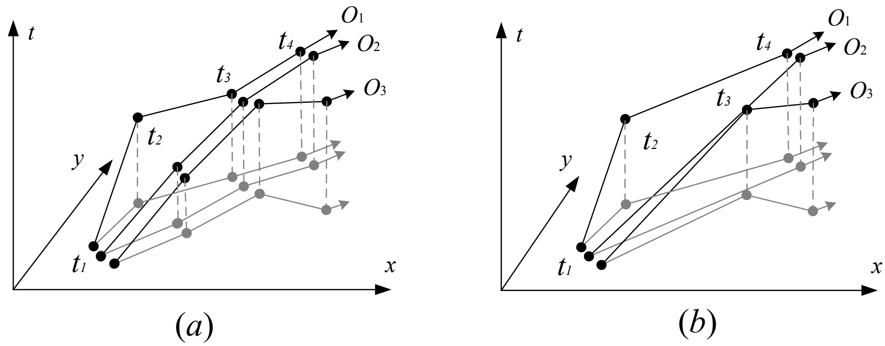


Fig. 5.18 An example of trajectory simplification.

5.6.2.2 Indexing

Indexing is a popular approach to accelerate the discovery process in data mining. Here we present a couple of examples used in the discovery of the trajectory patterns presented earlier in this chapter.

Kalnis *et al.* [36] employ a grid index G_t at each time point t for storing the data points at that time. The density-based clustering algorithm DBSCAN [22] is then applied on the grid index G_t in order to identify the clusters at time t.

Gudmundsson *et al.* [27] utilize the quadtree [49], the compressed quadtree [7], and the skip-quadtree [21] for the fast discovery of the flock, leadership, convergence, and encounter patterns. Specifically, they first index the spatial data points at each time point using the compressed quadtree. They then position a disc with radius r into the non-empty entries of the index and enlarge the disc to have radius $(1+\delta)\cdot r$ when necessary. The relative motion patterns are then retrieved by repeating the above process at each time point, and comparing the objects in the disc with the others in the neighbor time points.

5.6.2.3 The Apriori Approach

The Apriori approach has been applied to discover trajectory patterns efficiently [57, 44]. First, some (short) candidate trajectory patterns are generated, and their frequencies are counted by scanning the trajectory database. Then unpromising patterns (with low frequencies) are removed from consideration. The remaining candidates are combined together to form larger candidate patterns. This process is repeated until no further candidates are generated. The Apriori approach is able to reduce the search space significantly in the first few iterations, allowing trajectory patterns to be mined efficiently.

5.7 Summary

This chapter offers an overview of trajectory pattern mining. As position data is increasingly becoming available from a variety of applications and in larger and larger quantities, the rapidly advancing field of data mining is gaining in importance. The chapter first identifies several important applications of trajectory pattern mining. The trajectory pattern mining has been studied extensively in the literature, and many different definitions of patterns are available. We provide a categorization of patterns and discuss their similarities and differences. We examine three types of representative patterns in detail, namely Relative Motion Patterns, Disc-Based Trajectory Patterns, and Density-Based Trajectory Patterns. In addition, we cover distance measures and data transformation methods, indexing methods, and mining techniques for the discovery of trajectory patterns.

References

1. Digital battle field. http://www.defenselink.mil/news/newsarticle.aspx?id=45084
2. Porcupine caribou herd satellite collar project. http://www.taiga.net/satellite/.
3. Al-Naymat, G., Chawla, S., Gudmundsson, J.: Dimensionality reduction for long duration and complex spatio-temporal queries. In: Proceedings of the ACM symposium on Applied computing, pp. 393–397 (2007)
4. Andersson, M., Gudmundsson, J., Laube, P., Wolle, T.: Reporting leadership patterns among trajectories. In: SAC, pp. 3–7 (2007)
5. Andersson, M., Gudmundsson, J., Laube, P., Wolle, T.: Reporting leaders and followers among trajectories of moving point objects. GeoInformatica **12**(4), 497–528 (2008)
6. Arumugam, S., Jermaine, C.: Closest-point-of-approach join for moving object histories. In: Proceedings of the IEEE International Conference on Data Engineering, p. 86 (2006)
7. Arya, S., Mount, D.M., Netanyahu, N.S., Silverman, R., Wu, A.Y.: An optimal algorithm for approximate nearest neighbor searching. In: SODA, pp. 573–582 (1994)
8. Aung, H.H., Tan, K.L.: Discovery of evolving convoys. In: Proceedings of the 22nd international conference on Scientific and statistical database management, pp. 196–213 (2010)
9. Bakalov, P., Hadjieleftheriou, M., Keogh, E., Tsotras, V.J.: Efficient trajectory joins using symbolic representations. In: Proceedings of the international conference on Mobile data management, pp. 86–93 (2005)
10. Bakalov, P., Hadjieleftheriou, M., Tsotras, V.J.: Time relaxed spatiotemporal trajectory joins. In: Proceedings of the ACM international symposium on Advances in geographic information systems, pp. 182–191 (2005)
11. Benkert, M., Gudmundsson, J., Hbner, F., Wolle, T.: Reporting flock patterns. Computational Geometry **41**(1), 111125 (2008)
12. Brilingaitė, A.: Location-related context in mobile services. Ph.D. dissertation, Aalborg University (2006)

13. Brinkhoff, T., Kriegel, H.P., Seeger, B.: Efficient processing of spatial joins using r-trees. In: SIGMOD Conference, pp. 237–246 (1993)
14. Cao, H., Mamoulis, N., Cheung, D.W.: Discovery of periodic patterns in spatiotemporal sequences. TJDE **19**, 453–467 (2007)
15. Chen, L., Ng, R.: On the marriage of lp-norms and edit distance. In: Proceedings of the international conference on Very large data bases, pp. 792–803 (2004)
16. Chen, L., Özsu, M.T., Oria, V.: Robust and fast similarity search for moving object trajectories. In: Proceedings of the ACM SIGMOD international conference on Management of data, pp. 491–502 (2005)
17. Chen, Y., Patel, J.M.: Design and evaluation of trajectory join algorithms. In: GIS, pp. 266–275 (2009)
18. Chen, Z., Shen, H.T., Zhou, X.: Discovering popular routes from trajectories. In: ICDE, pp. 900–911 (2011)
19. Dodge, S., Weibel, R., Lautenschütz, A.K.: Towards a taxonomy of movement patterns. Information Visualization **7**, 240–252 (2008)
20. Douglas, D., Peucker, T.K.: Algorithms for the reduction of the number of points required to represent a line or its character. The American Cartographer **10**(42), 112–123 (1973)
21. Eppstein, D., Goodrich, M.T., Sun, J.Z.: The skip quadtree: a simple dynamic data structure for multidimensional data. In: SCG, pp. 296–305 (2005)
22. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 226–231 (1996)
23. Frank, A., Raper, J., Cheylan, J.P.: Life and motion of spatial socio-economic units. Taylor & Francis, London (2001)
24. Gaffney, S., Smyth, P.: Trajectory clustering with mixtures of regression models. In: KDD, pp. 63–72 (1999)
25. Gidófalvi, G., Pedersen, T.B.: Cab-sharing: An effective, door-to-door, on-demand transportation service. In: Proceedings of the 6th European Congress on Intelligent Transport Systems and Services (2007)
26. Gudmundsson, J., van Kreveld, M.: Computing longest duration flocks in trajectory data. In: Proceedings of the ACM international symposium on Advances in geographic information systems, pp. 35–42 (2006)
27. Gudmundsson, J., van Kreveld, M., Speckmann, B.: Efficient detection of motion patterns in spatio-temporal data sets. In: Proceedings of the ACM international symposium on Advances in geographic information systems, pp. 250–257 (2004)
28. Hadjieleftheriou, M., Kollios, G., Bakalov, P., Tsotras, V.J.: Complex spatio-temporal pattern queries. In: VLDB, pp. 877–888 (2005)
29. Han, J., Li, Z., Tang, L.A.: Mining moving object, trajectory and traffic data. In: DASFAA, pp. 485–486 (2010)
30. Iwase, S., Saito, H.: Tracking soccer player using multiple views. In: Proceedings of the IAPR Workshop on Machine Vision Applications (2002)
31. Jeong, S.H., Paton, N.W., Fernandes, A.A., Griffiths, T.: An experimental performance evaluation of spatiotemporal join strategies. Transactions in GIS **9**(2), 129–156 (2005)
32. Jeung, H., Liu, Q., Shen, H.T., Zhou, X.: A hybrid prediction model for moving objects. In: ICDE, pp. 70–79 (2008)
33. Jeung, H., Shen, H.T., Zhou, X.: Convoy queries in spatio-temporal databases. In: Proceedings of the IEEE International Conference on Data Engineering, pp. 1457–1459 (2008)
34. Jeung, H., Yiu, M.L., Zhou, X., Jensen, C.S.: Path prediction and predictive range querying in road network databases. The VLDB Journal **19**(4), 585–602 (2010)
35. Jeung, H., Yiu, M.L., Zhou, X., Jensen, C.S., Shen, H.T.: Discovery of convoys in trajectory databases. Proceedings of the VLDB Endowment **1**(1), 1068–1080 (2008)
36. Kalnis, P., Mamoulis, N., Bakiras, S.: On discovering moving clusters in spatio-temporal data. In: Proceedings of the International Symposium on Spatial and Temporal Databases, pp. 364–381 (2005)

37. Karimi, H.A., Liu, X.: A predictive location model for location-based services. In: Proceedings of the ACM International Symposium on Advances in Geographic Information Systems, pp. 126–133 (2003)
38. Laube, P., Imfeld, S.: Analyzing relative motion within groups of trackable moving point objects. In: GIScience, pp. 132–144 (2002)
39. Laube, P., Imfeld, S., Weibel, R.: Discovering relative motion patterns in groups of moving point objects. International Journal of Geographical Information Science **19**(6), 639–668 (2005)
40. Laube, P., van Kreveld, M., Imfeld, S.: Finding remo - detecting relative motion patterns in geospatial lifelines. In: Proceedings of the International Symposium on Spatial Data Handling, pp. 201–214 (2004)
41. Laube, P., Purves, R.S.: An approach to evaluating motion pattern detection techniques in spatio-temporal data. Computers, Environment and Urban Systems **30**(3), 347–374 (2006)
42. Lee, J., Han, J., Whang, K.: Trajectory clustering: a partition-and-group framework. In: Proceedings of the ACM SIGMOD international conference on Management of data, pp. 593–604 (2007)
43. Lee, J.G., Han, J., Li, X., Gonzalez, H.: *TraClass*: trajectory classification using hierarchical region-based and trajectory-based clustering. PVLDB **1**(1), 1081–1094 (2008)
44. Li, Z., Ding, B., Han, J., Kays, R.: Swarm: mining relaxed temporal moving object clusters. PVLDB **3**, 723–734 (2010)
45. Li, Z., Ding, B., Han, J., Kays, R., Nye, P.: Mining periodic behaviors for moving objects. In: SIGKDD, pp. 1099–1108 (2010)
46. Li, Z., Ji, M., Lee, J.G., Tang, L.A., Yu, Y., Han, J., Kays, R.: MoveMine: Mining moving object databases. In: Proceedings of the ACM SIGMOD international conference on Management of data, pp. 1203–1206 (2010)
47. Mamoulis, N., Cao, H., Kollios, G., Hadjieleftheriou, M., Tao, Y., Cheung, D.W.: Mining, indexing, and querying historical spatiotemporal data. In: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 236–245 (2004)
48. Sakr, M.A., Shams, A.: Spatiotemporal pattern queries. Geoinformatica **14** (2010)
49. Samet, H.: Foundations of Multidimensional and Metric Data Structures (The Morgan Kaufmann Series in Computer Graphics and Geometric Modeling). Morgan Kaufmann Publishers Inc. (2005)
50. Sumpter, N., Bulpitt, A.: Learning spatio-temporal patterns for predicting object behaviour. Image Vision Computing **18**, 697V704 (2000)
51. Tao, Y., Faloutsos, C., Papadias, D., Liu, B.: Prediction and indexing of moving objects with unknown motion patterns. In: Proceedings of the ACM SIGMOD International Conference on Management of Data, pp. 611–622 (2004)
52. Tao, Y., Papadias, D., Sun, J.: The tpr*-tree: An optimized spatio-temporal access method for predictive queries. In: Proceedings of the International Conference on Very Large Data Bases, pp. 790–801 (2003)
53. Tao, Y., Sun, J., Papadias, D.: Analysis of predictive spatio-temporal queries. ACM Transaction on Database Systems **28**(4), 295–336 (2003)
54. Vieira, M.R., Bakalov, P., Tsotras, V.J.: On-line discovery of flock patterns in spatio-temporal data. In: Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, pp. 286–295 (2009)
55. Vlachos, M., Gunopoulos, D., Kollios, G.: Discovering similar multidimensional trajectories. In: Proceedings of the IEEE International Conference on Data Engineering, pp. 673–684 (2002)
56. Šaltenis, S., Jensen, C.S., Leutenegger, S.T., Lopez, M.A.: Indexing the positions of continuously moving objects. In: Proceedings of the ACM SIGMOD International Conference on Management of Data, pp. 331–342 (2000)
57. Wang, Y., Lim, E.P., Hwang, S.Y.: Efficient mining of group patterns from user movement data. DKE **57**, 240–282 (2006)

58. Yi, B.K., Jagadish, H.V., Faloutsos, C.: Efficient retrieval of similar time sequences under time warping. In: Proceedings of the IEEE International Conference on Data Engineering, pp. 201–208 (1998)
59. Yoon, H., Shahabi, C.: Accurate discovery of valid convoys from moving object trajectories. In: IEEE International Conference on Data Mining Workshops, pp. 636–643 (2009)
60. Zheng, K., Trajcevski, G., Zhou, X., Scheuermann, P.: Probabilistic range queries for uncertain trajectories on road networks. In: EDBT, pp. 283–294 (2011)
61. Zhou, P., Zhang, D., Salzberg, B., Cooperman, G., Kollios, G.: Close pair queries in moving object databases. In: Proceedings of the ACM international symposium on Advances in geographic information systems, pp. 2–11 (2005)