Segmentation techniques for the summarization of individual mobility data

Maria Luisa Damiani*, Fatima Hachem*

Article Type:

Overview

Abstract

Segmentation techniques partition a sequence of data points in a series of disjoint sub-sequences - *segments* - based on some criteria. Depending on the context and the nature of data points, segments can be given an approximated representation. The final result is a *summarized* representation of the sequence. This intuitive mechanism has been extensively studied, for example, for the summarization of time series in order to preserve the 'shape' of the sequence while omitting irrelevant details. This survey focuses on the use of segmentation methods for extracting behavioral information from individual mobility data, in particular from spatial trajectories. Such information can then be given a compact representation in the form of summarized trajectories, e.g., *semantic* trajectories, *symbolic* trajectories. Two major streams of research are discussed, spanning computational geometry and data mining respectively, that are emblematic of the multiplicity of views.

*Department of Computer Science, University of Milan, Milan, Italy

INTRODUCTION

Data summarization is a major data mining task that can be concisely defined as *compressing data into an informative representation* [9]. That is, summarization discards irrelevant details from data while retaining the most important information. Unlike mere data compression, summarization involves the capability of abstracting content from data. Producing summaries of single or multiple documents is the most popular and intuitive application of the concept [17]. Other applications include the summarization of multimedia data [14], graphs [35], and time series [15]. In the light of the challenges posed by *big data*, data summarization represents an enabling factor for a more effective handling of very large volumes of complex information.

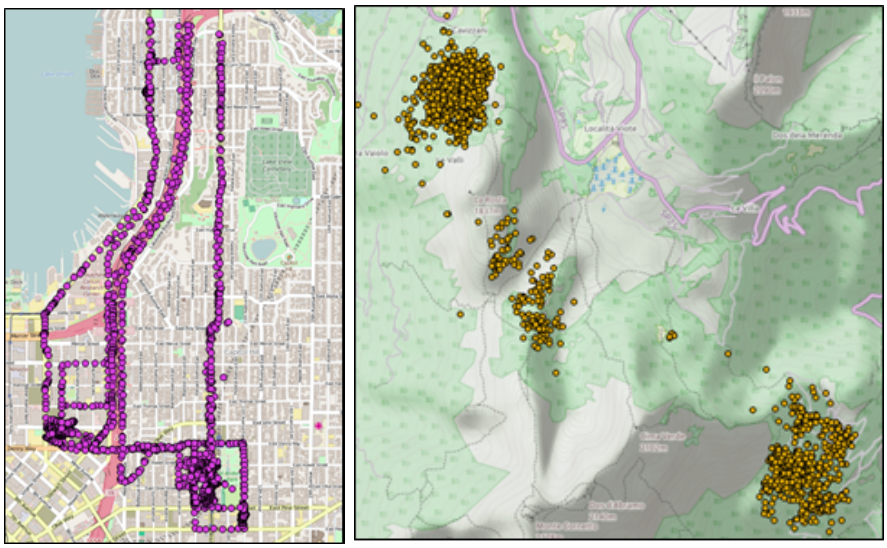
This article focuses on the summarization of mobility data. Broadly, mobility data regards the movements of objects in a reference space. Objects can represent entities of very different nature and scale, while the spaces of interest can be of arbitrary size. Moreover data can be dramatically voluminous, complex and heterogeneous. In the urban domain, for example, the data on the movement of vehicles possibly combined with additional contextual data, e.g. time series of environmental data, can help identify patterns of interest and/or make predictions, for example on relevant social and natural events, e.g. [10].

Very often mobility data take the form of *spatial trajectories*, namely sequences of coordinated locations reported at consecutive time instants and sampling the object movements in a time interval [39]. In more rigorous terms, given a reference space S, e.g. the Euclidean plane, a spatial trajectory T of length n is a sequence of n spatio-temporal *points*, i.e. $T = \{(p_i, t_i)\}_{i \in [1,n]}$ with $p_i \in S$ and $t_i < t_{i+1} \in Time$. Depending on the application, consecutive points can be equally spaced in time or not. Figure 1 shows two examples of spatial trajectories, reported on a planar map as sets of points, describing the movement of two different objects sampled at different frequencies.

Spatial trajectories summarization

The technological advances in positioning, communication and application services, have dramatically fostered the collection of massive volumes of spatial trajectories. Spatial tra-

jectories are, however, complex structured data, difficult to handle efficiently. Even conceptually simple operations such as *range* and *spatio-temporal join* queries are computationally costly, despite the indexing and query processing techniques employed in modern trajectory databases, e.g. [11, 20]. As a result, the efficient access and effective utilization of trajectory data remains an issue.



(a) (b)

Figure 1: (a) The movement of a person tracked for a few days at high sampling rate (by courtesy of John Krumm, [29]). (b) The movement of an animal (roe deer) tracked for over one year at low and irregular sampling rate [12]

To address such a problem, a possible strategy is to extract relevant time-dependent information from each individual movement and then use such coarser information in place of spatial trajectories. This strategy is especially appealing when the knowledge granule is not the spatio-temporal point itself, but rather the behavior exhibited by the individual in time. If such information can be extracted and then encoded into a compact form, i.e., as a summarized trajectory, then the dataset is much smaller and the access to relevant information potentially simpler. Along this line, a notion that has become popular in recent times is that of *semantic trajectory* [32]. Semantic trajectory has been proposed for the representation of time-varying behavioral information on single individuals. More recent proposals include *symbolic trajectories* [21] and *spatio-textual trajectories* [23]. A common

feature of all of these models is that they provide a rich representation of the movement that goes beyond the spatio-temporal characterization. In that sense, such models can be seen as instances of the general concept of summarized trajectory.

This survey focuses on key techniques for the construction of summarized trajectories, independently from the data model chosen for their representation. In particular, the focus is on the key paradigm of trajectory segmentation [38]. The segmentation task partitions a trajectory into a series of subsequences - the segments - that are somehow homogeneous with respect to certain properties. The systematic use of segmentation for trajectory data summarization and representation has been first proposed in [36] as part of a methodological framework aiming at supporting the semantic trajectory discovery process. In this article we overview major directions of recent research and highlight the methodological differences among these directions. While several surveys on collective movement analysis have been published, e.g. [27], to our knowledge, this is first survey explicitly focusing on the individual movement.

Outline

The remainder of the article is structured as follows. Section 2 discusses in more detail the reference context and introduces two main categories of segmentation techniques, called *attribute-driven* and *pattern-driven*, respectively. Representative techniques for the two categories are next discussed in Section 3 and Section 4, respectively. In order to put the presentation into context, both these sections contain some preliminary background information on prior work. Open issues and trends are finally discussed in the conclusive Section 5.

SETTING THE CONTEXT

Trajectories handling: database vs. knowledge discovery

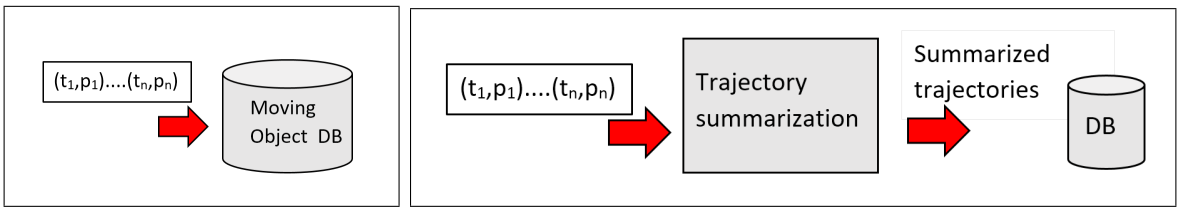
We start discussing in more detail the research context and the challenges that such a context poses. As emphasized earlier, spatial trajectories are complex to handle. As the movement of

an object in space is sampled for long periods and/or at high sampling frequency, the length of a spatial trajectory dramatically increases. Therefore, for large populations of moving objects, the amount of data becomes overwhelming.

One way to deal with large amounts of spatial trajectories is to store such data in a powerful database and use the functionalities of the system to access the data of interest. This is the *database-centric* view. The Moving Object data model is the reference paradigm for the management of databases of spatial trajectories [19]. In essence, a Moving Object database is a database equipped with a set of data types for the representation of spatial trajectories and their efficient manipulation through the use of dedicated operations and spatio-temporal indexes. It remains the fact that efficiency is still an issue while the deployment of this class of technologies, strictly rooted in the notion of spatial trajectory, is still limited. More recently, a number of distributed platforms for the management of big spatio-temporal data have been proposed, e.g. [2].

Opposed to the database-centric view is the *knowledge discovery view*. This different perspective finds a motivation in the fact that the availability of large amounts of trajectories, though complicating data processing, offers the opportunity of extracting valuable information on the time-evolving behavior of the involved objects. In general, the longer the trajectories (and the number of objects), the richer and more accurate the behavioral information that can be extracted from such data, for example on individual mobility patterns. As an example, consider the case of a geo-social application continuously sampling the location of a large number of individuals equipped with GPS enabled devices, e.g. smart-phones The amount of data that is continuously collected is huge. An approach to summarize such trajectories is to report only the sequence of places visited by each individual, e.g. working places, restaurants and so on, along with temporal information on the time spent in each of these places. As a result, the information relevant for the application, for example a user profiling application, is preserved while irrelevant details on the position occupied at a certain time can be omitted or handled separately. A practical application of this concept of trajectory summarization, in which the spatial trajectory is replaced by a sequence of places, can be found in *Google Maps Timeline*, an information service providing registered users with a map representation of the data collected by Google on their personal movement.

Figure 2 provides a simple visual summary of the data-centric view and knowledge discovery view.



(a) (b)

Figure 2: (a) Data-centric view: spatial trajectories are stored and accessed through a Moving Object database; b) Knowledge discovery view: spatial trajectories are first summarized, next possibly stored in some database or manipulated through some other application.

Segmentation techniques for the summarization of trajectories

A segmentation is a partition of a set of objects in a number of homogeneous parts. In general, when applied to spatial trajectories, the segmentation task can regard either the set of full trajectories or the points forming a single trajectory. The latter case is the topic of this article and thus we limit ourselves to consider this case.

More specifically, the segmentation of a spatial trajectory $T = \{(p_i, t_i)\}_i$ is a series of temporally ordered sub-trajectories, i.e.:

$$S_1 <_t .. <_t S_k \text{ with } T = \bigcup_{i=1}^{k} S_i$$

where $<_t$ denotes the relation of temporal ordering between sub-trajectories. The first point of every segment is called *breakpoint*. Segments can be annotated, for example with a label indicating the characterizing feature of the segment, or be associated with a representative point. Figure 3 shows an example of segmentation of a spatial trajectory consisting of labeled segments.

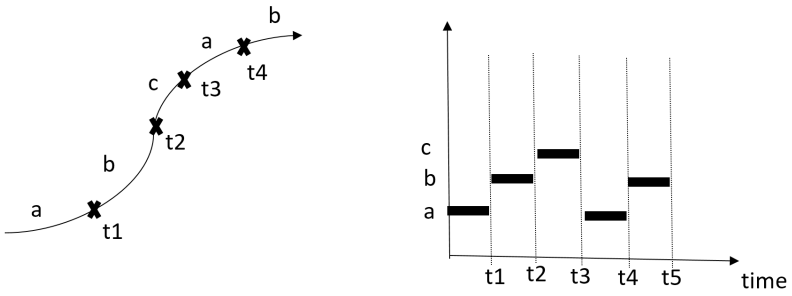


Figure 3: Segmentation of a spatial trajectory: the breakpoints along the input trajectory identify the begin/end of segments. Every segment corresponds to some property that holds for the whole duration of the segment

Abstractly, the segmentation task on a trajectory T can be expressed as a function $f(T, C) = \{S_1, .., S_k\}$ where C is the homogeneity criterion and $S_1, .., S_k$ are the k segments (or breakpoints) in which the trajectory is split. Often, the number k of segments is not known in advance. Key notion in segmentation is that of homogeneity criterion. There are two main interpretations of this concept of homogeneity:

- Homogeneity is defined with respect to selected properties of the movement (*movement attributes*) that can be derived from the geometric properties of spatial trajectories. Movement attributes are, for example, speed, heading, curvature. A segment is homogeneous if the conditions specified on such attributes are satisfied by the points of the segment. We refer to the class of segmentation methods which rely on this notion of homogeneity as *attribute-driven*.

- Homogeneity is defined with respect to the activity performed by the object in the corresponding time interval. We can think of activities as behavioral patterns that can be extracted from spatial trajectories using knowledge discovery techniques. Example activities are residing in a region or migrating from one residence to another residence. A segment is thus homogeneous if it can be associated with a certain activity. We refer to the corresponding class of segmentation techniques as *pattern-driven*.

The taxonomy in Figure 4 reports the general classification of the segmentation techniques. In the following, we will explore attribute-driven and pattern-driven segmentation

techniques..

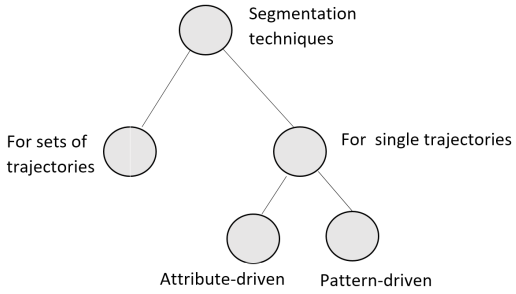


Figure 4: General taxonomy for segmentation techniques

ATTRIBUTE-DRIVEN SEGMENTATION

This class of techniques has its roots mainly in computational geometry [18, 3], though also inspired by work on time series. To highlight similarities and differences with prior work, we first provide some background on time series segmentation.

Time series segmentation

A time series is an arbitrary long sequence of correlated numeric values $r_1, .., r_n$ with $r_i \in \mathcal{R}$ typically collected from measurements made at uniformly spaced time instants [15], e.g. the series of temperature measurement over a period of time. A time series of length n is commonly modeled as a point in a n-dimensional space [25]. Therefore an operation such as the Euclidean distance between two time series of equal length $T = r_1, .., r_n$ and $T' = r'_1, .., r'_n$ can be expressed as the distance between two n-dimensional points. Unfortunately, distance-based operations, such as similarity-based search, performed over a large number of long sequences, can be extremely inefficient. Moreover the notion itself of distance in a high-dimensional space is problematic because of the curse of dimensionality [4]. Therefore segmentation is often recognized as a key approach to dimensionality reduction [25]. The segmentation task splits a time series in a number of sub-sequences and replaces each sub-sequence with an approximated representation based on a *segment model* [26]. For example, a segment can be represented by a single numeric value, e.g. a median data point in the

sub-sequence or by a polynomial or higher order curve.

The *segmentation algorithm* determines the breakpoints along the sequence based on the input parameters, typically the number of segments to be searched for. The problem of finding the breakpoints in the sequence is commonly framed as an optimization problem, specifically, given a time series of length n compute $k << n$ segments so as to minimize the error based on some error function. The error function can be for example the Euclidean distance from the approximating curve or point [34].

Spatial trajectories vs. time series. In the data mining literature, spatial trajectories are often equated to multivariate time series, namely the X-coordinate and Y-coordinate of the spatial trajectory form the components of the multivariate series [28]. However, the two notions of time series and spatial trajectories are slightly different especially with respect to the role of time and segmentation. In particular, the key feature of time series is the sequential structure of correlated data, while the time attribute is simply a property inducing a total order over a discrete set of values. Moreover, the goal of the segmentation task is to reduce the length of the time series to a predefined number of data points. By contrast, in spatial trajectories, the temporal information is the basis for the distinction between *discrete* and *continuous* movement. Whenever the movement is continuous, the missing points between two consecutive samples can be estimated by interpolation and a number of additional properties can be computed with a certain accuracy, such as trajectory curvature, speed and velocity. That is, a spatial trajectory can be seen as a sequence of data points with associated a number of time-varying scalar and vector functions. Such properties are those utilized by the techniques for attribute-based segmentation, discussed next.

Attribute-driven segmentation of spatial trajectories

The problem can be informally formulated as the problem of partitioning a spatial trajectory in a *minimum number of segments* in such a way that the movement inside each segment is nearly uniform with respect to some condition on movement attributes (called segmentation criteria, hereinafter). The number of segments is commonly unspecified, while the segmentation criteria can be grouped in three classes: *monotone, non-monotone* and *appli-*

cation specific. These criteria are described in what follows. The extended taxonomy for segmentation techniques is shown in Figure 5.

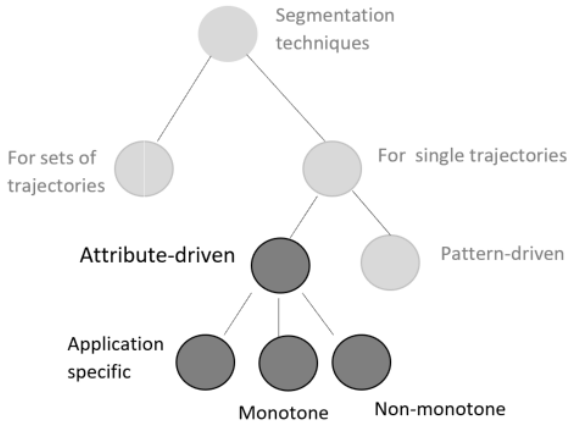


Figure 5: Taxonomy refinement: attribute-driven segmentation

We term 'application specific' those segmentation criteria that are defined on ad-hoc basis. For example, in [37], the problem is to split a spatial trajectory in a minimum number of segments approximating a linear movement at a constant speed.

More ambitious is the goal of defining a more general model grounded on rigorously defined concepts and providing theoretical guarantees. The first attempt in this direction is centered on the notion of *monotone criterion* [6]. A criterion is defined as a constraint on a movement attribute that has to be satisfied by all of the points in the segment. For example, a criterion can require the difference Δ_{speed} between the maximum and minimum speed value in the segment to be lower than $50km/h$. A criterion is *monotone* if for any sub-trajectory τ, it holds that if τ satisfies the criterion, then any sub-trajectory $\tau' \subseteq \tau$ also satisfies the criterion. For example the criterion $\Delta_{speed} < 50k/h$ is monotone while $\Delta_{speed} > 50k/h$ is not. Multiple criteria can be combined to form a set of criteria. It is shown that given a set of monotone criteria the optimal segmentation satisfying such criteria can be computed efficiently, in nearly nearly linear time with respect to the trajectory length.

We mention an interesting experience of application/evaluation of such a segmentation framework that also highlights important limitations of the approach [7]. The case study regards the movement of a group of birds monitored during their Spring migration from Netherlands to Siberia. The ultimate goal of the application is to discriminate the birds

activities, called *states*, such as flying and resting using a proper set of segmentation criteria. For example, the status of *flight* is informally defined as 'little variation in heading, speed at least 20 km/h for at least 5 hours'. Unfortunately, the constraint on the minimum time duration (e.g., at least 5 hours) is not monotone. Moreover, real data often contain outliers that cause incorrect breakpoints. That is, there exist segmentation criteria of practical interest that cannot be expressed using the proposed framework.

The third and more recent class of segmentation criteria tries to overcome the above issues by extending the theoretical framework to encompass non-monotone criteria. [3]. It is shown that computing the minimum number of segments satisfying non-monotone criteria is computationally hard under the assumption of continuous movement. Under certain circumstances, however, and only for certain non-monotone criteria, it is shown that the optimal segmentation can be computed efficiently in polynomial-time. In particular two criteria are found for which the problem is tractable. One such criterion requires that the difference between the maximum and minimum value of an attribute inside the segment does not exceed a given value, while allowing a certain percentage of outliers. The second criteria requires that on each segment the standard deviation of the attribute is below a certain threshold [3]. Interestingly, both these criteria are related to noise handling. However, a complete characterization of the non-monotone criteria for which efficient segmentation methods can be found is still open.

PATTERN-DRIVEN SEGMENTATION TECHNIQUES

This second class of techniques is based on machine learning methods. This is a broad category, with a strong application flavor. For example, the segmentation techniques commonly employed for transportation mode detection often involve the use of supervised learning methods [38]. To limit the scope of the survey to a manageable size, we restrict the focus on segmentation methods relying on unsupervised methods (clustering). Commonly, segmentation methods based on clustering are employed for the detection of specific behaviors (or patterns). Such patterns can be either defined with respect to specific domains, e.g. human mobility, or be generic, such as the *stop-and-move* pattern. In particular, a stop-and-move

pattern is an abstraction of the mobility behavior of an object that repeatedly stays for some time in a small region (i.e. a stop) before moving to some other regions. Objects exhibiting such a behavior include for example animals tracked while foraging or migrating, and the eyes gaze exploring a visual scene. Before proceeding, we provide some background information on clustering techniques for the aspects relevant for the discussion.

Background on clustering of temporally annotated data

The clustering task subdivides a set of objects in groups of similar objects based on some criteria of similarity. A broad range of clustering techniques have been proposed, based on diverse paradigms such as partitional, hierarchical, density-based, grid-based clustering [22]. Classically, all of these methods apply to unordered sets. Two popular techniques are conceptually relevant for the trajectory segmentation problem, *K-means* and *DBSCAN*. *K-means* is representative of the class of partitional clustering techniques. Accordingly, the clustering problem is to find a partition of k clusters, with k input parameter, that optimizes a properly defined clustering quality function. In contrast, *DBSCAN* generates clusters based on density criteria. The number of clusters is unknown, while the two input parameters ϵ and N specify the density requirements, i.e. ϵ, the radius of a neighborhood, and N the minimum number of data points in a ϵ-neighborhood, respectively [16]. Unlike *K-means*, *DBSCAN* is robust with respect to noise, therefore the presence of unstructured points does not have a disruptive impact on the partitioning of the set as in *K-means*.

Time-aware clustering. An important class of techniques addresses the problem of clustering temporally annotated data points. Two major categories of such techniques regard the clustering of spatio-temporal events, and the clustering of stream data, respectively. In particular: *(i) Spatio-temporal events* are uncorrelated spatio-temporal points, e.g. seismic events. The goal of the clustering task is to group together the events that are close in space and time. This problem is commonly approached introducing some notion of spatio-temporal distance. For example, *ST-DBSCAN*, a temporal extension of *DBSCAN*, introduces two metrics, one for space and one for time [5]. Accordingly, the key notion of ϵ-neighborhood for a point p is slightly modified, i.e. the ϵ-neighborhood contains N points whose temporal dis-

tance from p does not exceed a threshold value. *(ii) Data streams* are unbounded sequences of data of arbitrary type. The clustering task groups the data points as they arrive, under the memory constraints imposed by the data streaming context. In this case, the temporal information can be utilized, for example, to limit the amount of data to cluster, e.g. the clusters in the last year, last month, last week, as in [1]. A different strategy is to use the temporal information to award recent and evolving clusters against oldest and stable clusters. For example, in *DensStream* [8] every point is given a weight that decreases exponentially with time via a fading function $f(t) = 2^{-\lambda t}$, where $\lambda > 0$ is a system defined parameter. As the cumulative weight of a dense group of points exceeds a threshold value then such a group becomes a cluster. Next, if no new point is added, the weight will decay gradually until the group of points becomes noise and eventually the memory space is released for new clusters.

A common feature of all of these techniques is that the resulting clusters are not temporally separated. Therefore such methods are conceptually unsuitable for the segmentation of spatial trajectories.

Clustering techniques for pattern-driven segmentation

We now focus on clustering techniques that return temporally disjoint clusters. We refer to this form of segmentation as *clustering-based*. Another term often used is *sequential clustering* [30]. We distinguish three major classes of techniques: *heuristic-driven, density-based, partitional*. Accordingly, the general taxonomy introduced earlier can be refined as shown in Figure 6. An orthogonal and important distinction to bear in mind during the analysis of these categories is between the techniques that are sensitive to noise and those that are not. This distinction will be discussed at the end of this Section.

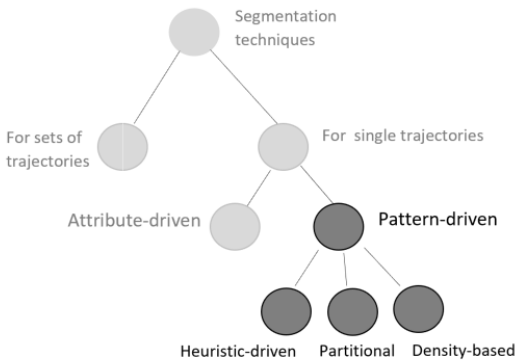


Figure 6: Taxonomy refinement: pattern-driven segmentation

Heuristic-driven segmentation. This class encompasses methods relying on simple heuristics. An early approach, which can be taken as representative of the class, is proposed in [24] for the analysis of human mobility. The application goal is to detect the places of interest visited by an individual, where a place of interest is a region where the individual stays for a minimum time. The idea is to compare every new point that arrives with the centroid of the current cluster. If far away from the centroid, the point is considered to belong to a different cluster. Finally, the clusters which represent places of interest are filtered out based on the duration threshold. Along this line, another popular technique has been proposed by Yu Zheng et al. [40]. A major problem with this class of approaches is the lack of a more general framework providing guarantees.

Based on partitional clustering. These approaches are inspired by *K-means*. For example, *Warped K-means* is an algorithm that, like *K-means*, allocates data points based on the analysis of the effects on a quality function, caused by moving a sample from its current cluster to a potentially better one [30]. Because of the ordering constraint, the first half of points in cluster j are only allowed to move to cluster $j-1$, and, respectively, the last half of data points are only allowed to move to cluster $j+1$. A point will be reallocated if and only if the operation is beneficial for the quality of clustering. This process is iterated until no transfers are performed. A more recent approach, taking inspiration from *Warp K-means*, employs the notion of density in place of the quality function, without requiring in input the parameter k [33]. In practice, a breakpoint is created as the density of the data

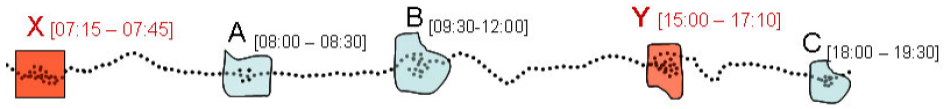
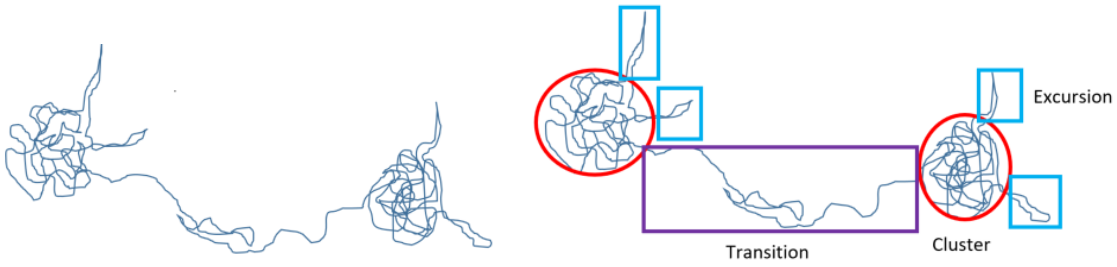


Figure 7: CB-SMoT clustering: the output is a sequence of temporally ordered stops. Every stop is associated with a time interval [31]

points in proximity of the current cluster representative is less than a threshold value. As relying on *K-means*, these techniques are sensitive to noise.

Based on density-based clustering. This class of approaches relies on *DBSCAN*. One of the earliest and probably most representative approaches is *CB-SMoT* (Clustering-Based Stops and Moves of Trajectories). This is a technique proposed for the extraction of *stops* from spatial trajectories. Stops are defined as segments of minimum length and minimum duration, along which the speed is thus limited [31]. An example from the original article and illustrating a sequence of stops is shown in Figure 7. *CB-SMoT* inherits key concepts from *DBSCAN*, yet such concepts are formulated in slightly different terms. In particular the notion of ϵ-neighborhood (neighborhood of radius ϵ) is defined along the linear representation of the trajectory, Moreover the constraint on the minimum number of points for a ϵ-neighborhood to be dense is reformulated as temporal constraint on the minimum duration of the sub-trajectory. Whenever the condition on the minimum speed for a minimum time is no longer satisfied, the cluster-segment is broken. This happens irrespective of the fact that the variation can be only temporary and thus could represent noise. That is, unlike *DBSCAN*, this technique is not robust against noise.

Noise handling. As we have seen, noise sensitivity is an issue common to all of those techniques. To deal with this problem, a common strategy is to introduce some additional constraints, for example on the maximum number of noise points that can be tolerated before a breakpoint is added, as in e.g. [24]. Such a parameter is, however, hard to set, especially whenever the sampling intervals are irregular and the clustering is to be applied to a large number of trajectories. As a result, these techniques are highly time consuming and little



(a) (b)

Figure 8: (a) Spatial trajectory; (b) SeqScan clustering: clusters, excursion points, transition points

effective in practice.

A first attempt to deal in more systematic way with the problem of noise in clustering-based segmentation is represented by *SeqScan* [12]. *SeqScan* is a clustering technique for the segmentation of sequences, fully compliant with the *DBSCAN* model. SeqScan distinguishes two classes of outliers: the points indicating a temporary absence from the cluster (*excursion point*) and the points representing a definitive departure from the cluster towards another cluster (*transition point*). The measure of *presence* is an estimate of the time spent inside the cluster excluding the periods of absence. The segmentation algorithm determines the sequence of clusters along with the classified outliers. For validation purposes, the approach has been experimentally used for the analysis of the migratory behavior of a group of animals [13]. Figure 8 shows the components of a segmentation, i.e. the clusters, the excursion points, and the transition points, for an example trajectory.

RESEARCH ISSUES AND CONCLUDING REMARKS

Two aspects deserve further discussion: (a) perspectives of the two research directions on segmentation techniques, and (b) the relationship between segmentation and representation of summarized trajectories. These aspects are discussed next.

(a) We have seen two major categories of approaches addressing the problem of segment-

ing spatial trajectories. While the potential applications for these techniques are similar, the methodologies underneath are substantially different. Both directions present pros and cons. Attribute-driven methodologies are built on solid theoretical frameworks. Yet, the characterization and generalization of the segmentation criteria, the handling of outliers and the scalability of the segmentation algorithms still represent important challenges. Probably less robust and general from a theoretical perspective but more flexible, with respect to the application needs, are the pattern-driven methodologies. In particular, cluster-based segmentation techniques are especially useful for the detection of *stop-and-move* patterns, a specific class of patterns that finds an application in a variety of domains. An open issue is the definition of validation methodologies specifically targeting this form of clustering over sequences. For example, an aspect that deserves some attention is whether the quality indexes commonly used for the evaluation of classical clustering can be straightforwardly applied on sequential clustering. The definition of theoretically robust frameworks possibly accounting for supplementary patterns is another major challenge.

(b) The second important aspect to consider concerns the representation of the segments resulting from the trajectory partitioning. Earlier in the paper, we have mentioned the notion of *semantic trajectory*, that is, a trajectory encompassing some form of knowledge on the individual movement. The notion of semantic trajectory has been given a more explicit characterization as sequence of *episodes* where an episode is substantially a segment annotated with application-dependent information [32]. A semantic trajectory is defined as the result of a process that starts from the acquisition of a raw sequence of spatio-temporal points, and goes through data cleaning, spatial trajectory segmentation and finally segment annotation. In this sense, segmentation techniques are instrumental to the generation of semantic trajectories. It remains the fact that the notion of semantic trajectory is exclusively defined at conceptual level, therefore is not given a workable specification. Recent work attempts to fill this gap. In particular, the model of *symbolic trajectories* [21] provides a way for representing segmented trajectories in a database. In such a model a segment is a pair *(TimeInterval, Label)* where *Label* is a text unit describing the individual behavior in the time interval. Symbolic trajectories are represented in a database as values of a properly

defined data type and queried using a pattern-based query language.

In summary, an emerging trend is to manage summarized trajectories through a database. We recall that we have started contrasting the database and the knowledge discovery view, as two orthogonal strategies for the effective management of large volumes of trajectory data. In the light of these last considerations, it appears that these two views are, in reality, converging towards the definition of a unifying framework enabling the efficient access to content-rich trajectory datasets and as an alternative to parallel and distributed spatio-temporal databases.

References

[1] Charu C. Aggarwal, Jiawei Han, Jianyong Wang, and Philip S. Yu. A framework for clustering evolving data streams. In *Proc. of the International Conference on Very Large Data Bases*, 2003.

[2] Louai Alarabi. ST-Hadoop: A MapReduce Framework for Big Spatio-temporal Data. In *Proc. of the ACM International Conference on Management of Data*, 2017.

[3] Boris Aronov, Anne Driemel, Marc Van Kreveld, Maarten Löffler, and Frank Staals. Segmentation of Trajectories on Nonmonotone Criteria. *ACM Trans. Algorithms*, 12 (2):1–28, 2015.

[4] Kevin Beyer, Jonathan Goldstein, Raghu Ramakrishnan, and Uri Shaft. *When Is "Nearest Neighbor" Meaningful?*, chapter in: International Conference on Database Theory (ICDT99). Lecture Notes in Computer Science. Springer, 1999.

[5] Derya Birant and Alp Kut. ST-DBSCAN: An algorithm for clustering spatial-temporal data. *Data & Knowledge Engineering*, 60(1):208 – 221, 2007.

[6] Maike Buchin, Anne Driemel, Marc J. van Kreveld, and Vera Sacristán Adinolfi. Segmenting trajectories: A framework and algorithms using spatiotemporal criteria. *J. Spatial Information Science*, 3:33–63, 2011.

[7] Maike Buchin, Helmut Kruckenberg, and Andrea Kölzsch. *Segmenting Trajectories by Movement States*, chapter Advances in Spatial Data Handling, pages 15–25. Springer, 2013.

[8] Feng Cao, Martin Ester, Weining Qian, and Aoying Zhou. Density-based clustering over an evolving data stream with noise. In *Proc. SIAM*, 2006.

[9] Varun Chandola and Vipin Kumar. Summarization – compressing data into an informative representation. *Knowledge and Information Systems*, 12(3):355–378, 2007.

[10] Fernando Chirigati, Harish Doraiswamy, Theodoros Damoulas, and Juliana Freire. Data polygamy: The many-many relationships among urban spatio-temporal data sets. In *Proc. of the International Conference on Management of Data*, SIGMOD '16, 2016.

[11] Philippe Cudré-Mauroux, Eugene Wu, and Samuel Madden. TrajStore: An adaptive storage system for very large trajectory data sets. In *Proc. of the International Conference on Data Engineering*, 2010.

[12] Maria Luisa Damiani, Hamza Issa, and Francesca Cagnacci. Extracting stay regions with uncertain boundaries from GPS trajectories: a case study in animal ecology. In *Proc. of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 2014.

[13] Maria Luisa Damiani, Hamza Issa, Giuseppe Fotino, Marco Heurich, and Francesca Cagnacci. Introducing 'presence' and 'stationarity index' to study partial migration patterns: an application of a spatio-temporal clustering technique. *International Journal on Geographical Information Science*, 30(5):907–928, 2016.

[14] Duo Ding, Florian Metze, Shourabh Rawat, Peter Franz Schulam, Susanne Burger, Ehsan Younessian, Lei Bao, Michael G. Christel, and Alexander Hauptmann. Beyond audio and video retrieval: Towards multimedia summarization. In *Proc. of the 2Nd ACM International Conference on Multimedia Retrieval*, 2012.

[15] Philippe Esling and Carlos Agon. Time-series data mining. *ACM Comput. Surv.*, 45 (1):1–34, 2012.

[16] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A Density-based Algorithm for Discovering Clusters a Density-based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In *Proc. of the International Conference on Knowledge Discovery and Data Mining*, 1996.

[17] Amir Gandomi and Murtaza Haider. Beyond the hype: Big data concepts, methods, and analytics. *International Journal of Information Management*, 35(2):137 – 144, 2015.

[18] Joachim Gudmundsson, Patrick Laube, and Thomas Wolle. Movement patterns in spatio-temporal data. In *Encyclopedia of GIS*, pages 726–732. 2008.

[19] R H. Güting, M.H. Böhlen, Martin Erwig, C.S. Jensen, N.A. Lorentzos, M. Schneider, and M.Vazirgiannis. A foundation for representing and querying moving objects. *ACM Transactions on Database Systems*, 25(1):1–42, 2000.

[20] Ralf H. Güting, Thomas Behr, and Christian Düntgen. *SECONDO: a Platform for Moving Objects Database Research and for Publishing and Integrating Research Implementations*. Fernuniv., Fak. fr Mathematik u. Informatik, 2010.

[21] R.H. Güting, F. Valdés, and M. L. Damiani. Symbolic trajectories. *ACM Transactions on Spatial Algorithms and Systems*, Vol. 1, Issue 2:7, 2015.

[22] Jiawei Han. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers Inc., 2005.

[23] Hamza Issa and Maria Luisa Damiani. Efficient access to temporally overlaying spatial and textual trajectories. In *IEEE International Conference on Mobile Data Management (MDM)*, 2016.

[24] Jong Hee Kang, William Welbourne, Benjamin Stewart, and Gaetano Borriello. Extracting places from traces of locations. In *Proc. of the ACM International Workshop on Wireless Mobile Applications and Services on WLAN Hotspots*, 2004.

[25] Eamonn Keogh, Kaushik Chakrabarti, Michael Pazzani, and Sharad Mehrotra. Dimensionality reduction for fast similarity search in large time series databases. *Knowledge and Information Systems*, 3(3):263–286, 2001.

[26] Eamonn J. Keogh, Selina Chu, David Hart, and Michael J. Pazzani. An Online Algorithm for Segmenting Time Series. In *Proc. of the IEEE International Conference on Data Mining*, 2001.

[27] Slava Kisilevich, Florian Mansmann, Mirco Nanni, and Salvatore Rinzivillo. *Spatio-temporal Clustering*, chapter in: Data Mining and Knowledge Discovery Handbook. Springer, 2010.

[28] Dimitrios Kotsakos, Goce Trajcevski, Dimitrios Gunopulos, and Charu C. Aggarwal. Time-series data clustering. In *Data Clustering: Algorithms and Applications*. CRC Press, 2013.

[29] John Krumm and Dany Rouhanam. Placer: semantic place labels from diary data. In *Proc. of the ACM International Joint Conference on Pervasive and Ubiquitous Computing*. 2013.

[30] Luis A. Leiva and Enrique Vidal. Warped k-means: an algorithm to cluster sequentially-distributed data. *Information Sciences*, 237:196–210, 2013.

[31] Andrey Tietbohl Palma, Vania Bogorny, Bart Kuijpers, and Luis Otavio Alvares. A clustering-based approach for discovering interesting places in trajectories. In *Proc. of the ACM Symposium on Applied Computing*, 2008.

[32] C. Parent, S. Spaccapietra, C. Renso, G. Andrienko, N. Andrienko, V. Bogorny, M. L. Damiani, A. Gkoulalas-Divanis, J. Macedo, N. Pelekis, Y. Theodoridis, and Z. Yan. Semantic trajectories modeling and analysis. *ACM Comput. Surv.*, 45(4):1–32, 2013.

[33] Yehezkel S. Resheff. Online Trajectory Segmentation and Summary With Applications to Visualization and Retrieval. In *Proc. IEEE International Conference on Big Data*, 2016.

[34] Evimaria Terzi and Panayiotis Tsaparas. Efficient algorithms for sequence segmentation. In *Proc. of the SIAM International Conference on Data Mining*, 2006.

[35] Yuanyuan Tian, Richard A. Hankins, and Jignesh M. Patel. Efficient Aggregation for Graph Summarization. In *Proc. of the ACM SIGMOD International Conference on Management of Data*, 2008.

[36] Zhixian Yan, Dipanjan Chakraborty, Christine Parent, Stefano Spaccapietra, and Karl Aberer. SeMiTri: A Framework for Semantic Annotation of Heterogeneous Trajectories. In *Proc. of the International Conference on Extending Database Technology (EDBT)*, 2011.

[37] Hyunjin Yoon and Cyrus Shahabi. Robust time-referenced segmentation of moving object trajectories. In *Proc. ICDM*, 2008.

[38] Yu Zheng. Trajectory Data Mining: An Overview. *ACM Trans. Intelligent Systems and Technologies*, 6(3):1–41, 2015.

[39] Yu Zheng and Xiaofang Zhou. *Computing with Spatial Trajectories*. Springer, 2011.

[40] Yu Zheng, Lizhu Zhang, Zhengxin Ma, Xing Xie, and Wei-Ying Ma. Recommending friends and locations based on individual location history. *ACM Trans. Web*, 5(1):1–44, 2011.