Extracting stay regions with uncertain boundaries from GPS trajectories: a case study in animal ecology

Maria Luisa Damiani & Hamza Issa
Dept. Computer Science
University of Milan, Milan, Italy
{damiani,issa}@di.unimi.it

Francesca Cagnacci
Dept. Biodiversity and Molecular Ecology
Fondazione Edmund Mach
S. Michele all'Adige, Italy
francesca.cagnacci@fmach.it

ABSTRACT

In this paper we present a time-aware, density-based clustering technique for the identification of stay regions in trajectories of low-sampling-rate GPS points, and its application to the study of animal migrations. A *stay region* is defined as a portion of space which generally does not designate a precise geographical entity and where an object is significantly present for a period of time, in spite of relatively short periods of absence. Stay regions can delimit for example the residence of animals, i.e. the home-range. The proposed technique enables the extraction of stay regions represented by dense and temporally disjoint sub-trajectories, through the specification of a small set of parameters related to density and presence. While this work takes inspiration from the field of animal ecology, we argue that the approach can be of more general concern and used in perspective in different domains, e.g. the study of human mobility over large temporal scales. We experiment with the approach on a case study, regarding the seasonal migration of a group of roe deer.

Categories and Subject Descriptors

H.2.8 [**Database management**]: Database applications—*Spatial databases and GIS*

General Terms

Algorithms, experimentation

Keywords

Mobility patterns, clustering, animal ecology

1. INTRODUCTION

With the advances in mobile technologies, the extraction of behavioral patterns from collections of geometric trajectories regarding e.g. people, animals and goods, has become a prominent research issue in a variety of disciplines such as geography, transportation science, biology, computer science [1, 2, 3, 4]. For example, in the field of animal ecology, modern animal telemetry and sensor networks (e.g. GPS receivers and other sensors mounted on devices deployed on animals, such as collars) enable the collection of content-rich, fine-grained trajectories, opening up new opportunities for the study of the animal behavior [5]. In particular this paper takes inspiration from the migration patterns of wild animals. Analyzing animal mobility offers a number of advantages over e.g. human mobility. For example privacy is not an issue, thus mobility data can be freely shared within the research community, moreover the solutions can be validated using field work knowledge and experience. This paves the way to the development of effective techniques and to their deployment in different domains. In this spirit, we present a generic problem, and develop a solution which is deployed and evaluated on a case study in animal ecology.

SIGSPATIAL'14, November 04 - 07 2014, Dallas/Fort Worth, TX, USA
Copyright 2014 ACM 978-1-4503-3131-9/14/11$15.00
http://dx.doi.org/10.1145/2666310.2666417.

1.1 Case study

The case study regards the extraction of the seasonal migrations of a number of roe deer (small ungulate species that can live in a variety of environments), equipped with a low-sampling-rate GPS collar and tracked for a period covering a few seasons. The animals of this species can either migrate or be stationary, a behavior known as *partial migration* [6]. Moreover, whenever an animal migrates, the migration takes place with modalities and times that - although respecting certain general patterns, e.g. seasonality - can vary from animal to animal. This means that every animal has its own migration behavior (i.e. the movement pattern is individual [1]). At very coarse level, the behavior can be seen as a stop-and-move pattern [7], i.e. the animal stays in a region for some time and then moves to some other region. In reality the behavior is more complex. The animals spend most of their time inside a *home-range*. The concept of home-range is key in animal ecology. A popular definition of home-range is that of "area traversed by the individual in its normal activities of food gathering, mating and caring for young" [8]. In reality, the concept does not have a univocal interpretation (the interested reader can refer to [9] for details). Occasionally the animals can make *excursions* outside the home-range and possibly stay for short periods in a different area before returning to the home-range. A *migration* is a transition from one home-range to another home-range. During the migration the animal can stop in small areas for a short time (*stopover*). An example of migration pattern, which combines all these concepts, is reported in Figure 1(a). Note that the migration behavior

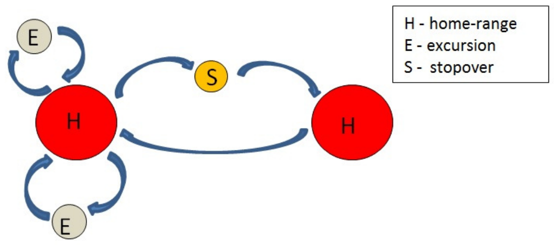


Figure 1: Example of animal migration pattern.

does not necessarily have a periodicity (e.g. animals may not migrate every year). Moreover the temporal and spatial extent of the regions in which the animals stay, as well as the duration of the moves can significantly vary [10]. As a matter of fact, there is no consensus in the scientific community on the fact that the distances covered by the animals of this species or their speed are good indicators of a migration in progress, while the collection of GPS traces, as for most middle to large vertebrate species, is a relatively recent practice, that needs to be still fully exploited in its potentialities. A remarkable initiative in this field is Eurodeer, a collaborative project started in 2009 and involving 29 European research institutes for the collection, organization and sharing of movement data, i.e. GPS, VHF and activity data, regarding over 900 roe deer in 25 areas in Europe[1]. The Eurodeer project [10] provides the application context for the research presented in this paper.

1.2 Requirements

We take inspiration from the case study to define a generic problem framework where: (i) we propose the concept of *stay region* to abstract away from the notion of animal's home-range, stopover and so on; (ii) the problem to address is to extract stay regions from low-sampling-rate GPS trajectories. In particular, we define *stay region* as a sequence of points spatially confined to a portion of space (not designating a geographical entity, e.g. forest), frequently visited by the object and in which the object spends sufficient time although experiencing periods of absence. No assumption is made on speed and other movement characteristics, as well on the distribution of points in stay regions.

A stay region is naturally an area that is dense of points. Density however is not sufficient to characterize a stay region, as an object is also requested to stay for sufficient time in the region. A straightforward interpretation of the notion of ”time elapsed in a region” is that of stay duration, the time difference between the last point and first point in the region. In reality, since the object can occasionally leave the stay region, the time effectively spent in the region is very likely shorter than the duration. Ideally, an object repeatedly moving back and forth between two areas, say A and B does not stay in any of the two regions, because there is no evidence that the individual remains for sufficient time in any of them. To capture this intuition we introduce the notion of *presence*. The presence in a stay region is the time plausibly elapsed in such a region. Plausibly means with reasonable evidence, given the uncertainty of the object location. Accordingly a stay region is a sub-trajectory defining a region which is dense and in which the object's presence is significant.

The problem to address can be finally formulated as follows. The goal is to extract the sequence of stay regions from a trajectory, considering that: (i) the stay regions are to be temporally disjoint, i.e. the temporal extent of a stay region does not overlap the temporal extent of another stay region. (ii) The number of stay regions in a trajectory is not known. (iii) Stay regions can have an arbitrary spatial shape while the temporal extent can be only coarsely predicted. (iv) Periods of presence and absence in a region interleave. (v) Mobility parameters, e.g. speed, direction, are not relevant to discriminate whether the object is inside or outside a stay region.

1.3 Approach and contribution

One could argue that the problem is not very different from detecting e.g. the points of interest visited by tourists such as in [3, 11]. For example Zheng et al. [3], defines a *stay point* as a set of consecutive GPS points of the trajectory, close to each other (based on distance threshold) and in which the user stays for a minimum time (duration threshold). The temporal scale at which the movement is observed is however a small scale (e.g. hours, days), moreover the semantics of the stay points is given by the geographical context. Conversely, in the case study the animals can stay in a region for months or years, and exhibit a complex behavior, such as moving back and forth from a region whose spatial and temporal boundaries are uncertain.

A different paradigm is trajectory segmentation [12, 13, 14, 15, 16]. Trajectory segmentation is extensively applied to extract stop-and-move patterns such as in [14, 13]. The idea of segmentation is to partition the trajectory in segments of maximal length where movement characteristics inside each segment are *monotone* i.e. the same characteristics hold in every subsegment of the segment [12]. Monotone characteristics include speed, heading, curviness. One could define, for example, a stay region as a segment is which the object's speed is below a threshold value. Unfortunately, there is no scientific or empirical evidence that these movement characteristics are really informative for the migration pattern we are considering (while it can be for specific migration patterns as in [17]). On a different front, density-based clustering is a popular and robust paradigm that has shown to be effective in various contexts including stream data analysis and data warehousing [18, 19, 20, 21]. It is worth noting that density-based clustering is also employed for the discovery of stops (i.e. low speed segments) in fine-grained human trajectories as in [22, 13]. The key idea behind those solutions is to extend the notion of distance to account of the temporal dimension (i.e. a cluster consists of points that are close both in space and time). We claim that such distance model is conceptually inadequate in our scenario. In fact, animals can experience periods of absence from the home range, thus subsequent points in the cluster are not necessarily close in time. In essence, constraining the temporal distance is not a solution, while we need to ensure that stay regions do not overlap in time while satisfying the presence requirements.

In this paper, we present a novel approach which combines the effectiveness of density-based clustering with the partitioning capability of trajectory segmentation without introducing any supplementary assumption on movement characteristics. The key idea is to only use density and presence

[1]http://www.eurodeer.org

as non-monotone criteria for the partitioning of a trajectory in a set of sub-trajectories of maximal length and temporally non-overlapping. A trajectory is subdivided in pieces alternating stay regions and non-stay regions. The key contributions of this paper can be summarized as follows:

- We introduce and define in a rigorous way the *stay regions discovery problem.*
- We develop a novel algorithm to extract the stay regions. The algorithm is grounded on the formal framework of DBSCAN [18]. The algorithm requires the specification of only three parameters. This facilitates the practical deployment of the technique. The algorithm is called SeqScan.
- We validate the algorithm on the case study, specifically regarding the behavior of a few tens of roe deer living in the same area, showing that the algorithm effectively detects migratory behavior previously described by means of statistics-based methods [10], and helps explaining uncertain cases. Moreover we illustrate a possible methodology of use.

The remainder of the paper is organized as follows: the research problem is formally defined in Section 2. Section 3 describes the algorithm. Section 4 presents the experimental part, mostly focused on the application of the technique on the case study. Section 6 overviews related research. The conclusive section discusses the plans for future work.

2. PROBLEM DEFINITION

Before defining the stay region discovery problem, we review the fundamental concepts of DBSCAN and provide a few basic definitions.

2.1 Background: the DBSCAN cluster model

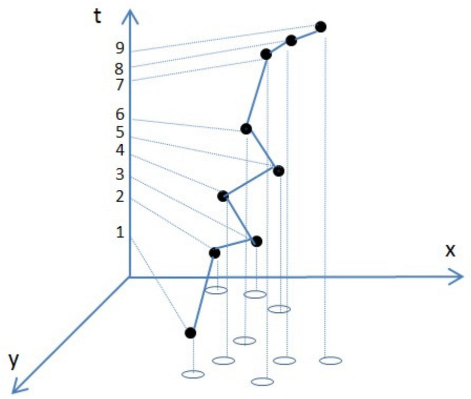
Consider a database P of points and the input parameters $\epsilon \in \mathbb{R}$ (i.e. the distance threshold), and $K \in \mathbb{N}$ (i.e. the minimum number of points that a cluster contains). Let $d()$ be the distance function.

The fundamental concepts of the DBSCAN cluster model are as follows [18]: (i) The *ϵ-neighborhood* of $p \in P$, denoted $N_\epsilon(p)$, is the subset of points that are "close" to p, i.e. $N_\epsilon(p) = \{p_i \in P, d(p, p_i) \leq \epsilon\}$. (ii) Point p is a *core point* if its ϵ-neighborhood contains at least K points, i.e. $|N_\epsilon(p)| \geq K$. A point that is not a core point but belongs to the neighborhood of a core point is a *border point.* (iii) Point p is *directly density-reachable* from q if q is a core point and $p \in N_\epsilon(q)$. (iv) Two points p and q are *density reachable* if there is a chain of points $p_1, .., p_n$, $p_1 = p, p_n = q$ such that p_{i+1} is directly reachable from p_i. (v) Points p and q are *density connected* if there exists a core point o such that both p and q are density-reachable by o. A cluster is finally defined as *a maximal set of density-connected points*, where maximal means that every point p that is reachable from a core point q belongs to the cluster containing q.

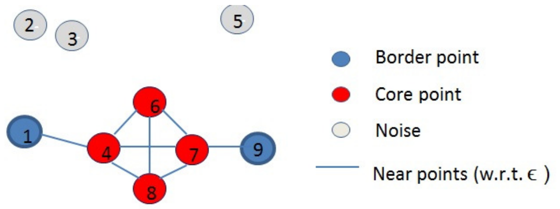
2.2 Preliminaries

Trajectory. A trajectory T is a sequence of spatio-temporal points $T = [p_1, .., p_n]$ with $p_i = (l_i, t_i)$ where l_i, t_i is the sampled location in space and time respectively with $t_i < t_{i+1}$ and n the length of the trajectory. The trajectory has a begin point p_{start}, an end point p_{end}, a temporal extent $[t_{start}, t_{end}]$, and a duration $|t_{start} - t_{end}|$. The duration is measured in e.g. days or in some other unit. No assumption is made on the sampling rate. However the distance in space covered by the object in the time interval $[t_i, t_{i+1}]$ is normally limited (in relation to the mobility pattern under consideration).

Sub-trajectory. A sub-trajectory $S = [p_i^1, .., p_j^m] \subseteq T$ of length m is a sequence of temporally ordered points of T with index $1, .., m$. A sub-trajectory may contain *gaps.* A gap is the open interval (t_i, t_j) signing a "hole" in the sequence, i.e. two points that are consecutive in S are not consecutive in T, i.e. $p_i^x, p_j^{x+1} \in S \rightarrow j \neq i+1$. For example, given $T = [p_1, .., p_9]$ the sub-trajectory $[p_3, p_5, p_8]$ contains two gaps, (t_3, t_5) and (t_5, t_8), respectively. The temporal extent of the sub-trajectory is $[t_3, t_8]$.



(a)



(b)

Figure 2: (a) Space-time cube of the example trajectory. (b) The points are projected on space. A subset of points forms a dense region with respect to $\epsilon, K = 3$

2.3 The stay region discovery problem

The notion of stay region is defined in terms of *density* and *presence.*

2.3.1 *Dense region*

A dense region in T is a sub-trajectory consisting of points that, projected on the plane, forms a DBSCAN cluster. Let $d(.)$ be the Euclidean distance. Formally:

DEFINITION 2.1 (DENSE REGION). *A dense region $S \subseteq T$ is a sub-trajectory $S = [q^1, .., q^m]$ such that the set of locations $[l^1, .., l^m]$ is a maximal density connected set with respect to ϵ and K. The points that do not belong to any dense region in T are qualified as noise.* $\diamond$

EXAMPLE 2.1. *Consider the trajectory $T = [p_1, .., p_9]$ illustrated in the space-time cube of Figure 2.1(a). The points projected on the plane are shown in Figure 2.1(b) and numbered following the temporal order. For the sake of readability, the pairs of points that are at distance less than or equal ϵ are connected by a line. The value of K is set to 3. It can be noticed that the sequence $S = [p_1, p_4, p_6, p_7, p_8, p_9]$ is a dense region.The points p_4, p_6, p_7, p_8 are core points while p_1, p_9 are border points. The region S contains two gaps,represented by the open intervals (t_1, t_4) and (t_4, t_6).*

2.3.2 Presence

A dense region S may contain gaps, where a gap indicates a period of absence from the region. We call *Time Segment* of a dense region the set of periods in which the object is present in such a region. A graphical representation of the Time Segment associated with the dense region in Example 2.1 is shown in Figure 3. This Time Segment consists of two instants and one period (the instant is a degenerated period where the two extremes coincide), i.e. $[t_1, t_1] \cup [t_4, t_4] \cup [t_6, t_9]$.

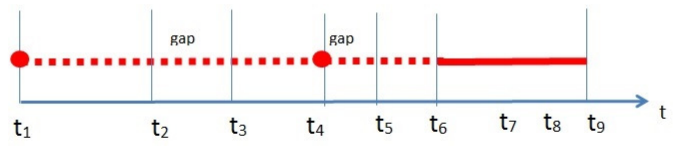


Figure 3: The Time Segment of the dense region in Example 2.1. The dotted lines indicate gaps. The circles correspond to instants.

Next, we introduce the notion of *presence* in incremental fashion. Let us start considering the case in which $p_i, p_{i+1} \in T$ are two consecutive points both included in the dense region S. A sensible question is whether the intermediate points falling in the interval $[t_i, t_{i+1}]$ belong to S or not. We know that the position is uncertain. Yet, we have assumed that the object's move between two subsequent samples is relatively limited in space. This legitimates the assumption that if $p_i, p_{i+1} \in S$ also the points in the interval $I = [t_i, t_{i+1}]$ are contained in S. Therefore the *presence* in I is computed as duration of I. Consider now the case in which only one of the points is in S, namely the object moves somewhere outside the area. In this case, it is plausible that the object is outside the dense region for most of the time. In this case, the presence in I is set to 0. By extension, we define the presence in S as the sum of the presence in each segment of S. A formal definition is given below:

DEFINITION 2.2 (PRESENCE). *Let $S = [q^1., .., q^m]$ be a dense region in $T = \{p_1, .., p_n\}$. Denote with $S[i, i+1]$ two consecutive points in S. We define:*

- *The presence in $S[i, i+1]$:*

$$P(S[i, i+1], T) = \begin{cases} |t_h - t_{h+1}|, & \text{if } \exists h, q^i = p_h, q^{i+1} = p_{h+1} \\ 0, & \text{otherwise} \end{cases}$$

- *The presence in the dense region S:*

$$P(S, T) = \sum_{i \in [1, n-1]} P(S[i, i+1], T)$$

- *We say that the presence in S is persistent with respect to $\delta > 0$ (*presence threshold*) if it holds: $P(S, T) > \delta$* ◇

The presence can be easily computed from the Time Segment. For example, given the Time Segment in Figure 3 the presence in the dense region is $|t_9 - t_6|$.

2.3.3 Problem formulation

DEFINITION 2.3 (STAY REGION). *A stay region S in the trajectory T is a sub-trajectory of T such that: (i) S is a dense region w.r.t. ϵ and K. (ii)The object's presence in S is persistent w.r.t δ*

Following the previous definition, the problem to address is to find the set of stay regions based on the values of the three parameters: ϵ, K, δ. Fundamental requirements are:(i) the stay regions are to be temporally disjoint. This means that a point in a stay region falling in the temporal extent of another stay region cannot exist. (ii) The stay regions ought to be of maximal length, that is if a point can be added to region S, then it belongs to S. The problem is summarized as follows:

DEFINITION 2.4 (STAY REGIONS DISCOVERY PROBLEM). *The problem is to extract from a trajectory T the sequence $[S_1, .., S_n]$ of temporally disjoint stay regions of maximal length. The points that do not fall in any stay region are* noise. *The subset of noise points temporally falling between the end of one stay region S_i and the beginning of the next one S_{i+1} form a* transition *(denoted $S_i \rightarrow S_{i+1}$).*

The set of stay regions and transitions determines a partition, i.e. *segmentation*, of the temporal extent of T.

3. THE ALGORITHM

A naive approach to the above problem is to apply the DBSCAN algorithm [18] over the set of points (projected over space) to find the set of dense regions and then determine those in which the presence is persistent. The drawback is that the resulting sub-trajectories are not temporally disjoint. Thus the solution does not work. A different approach is to scan sequentially the trajectory and progressively aggregate the points creating one stay region at a time. In this case, the question is how to recognize the end of a stay region. We recall, in fact, that no supplementary movement characteristic can indicate whether the object is inside or outside a stay region. To deal with this problem, we propose the following approach: (i) a stay region is seen as an attraction area, i.e. an area when the object returns after periods of absence. (ii) A stay region remains attractive for the object until a new stay region is found. This means that the boundaries of a stay region are only known when the next stay region is detected (or the trajectory terminates).

In the next, we refer to a stay region "in progress" with the term of *cluster* (not to be confused with the DBSCAN cluster). Clusters are dynamic entities with a life cycle. As illustrated in the state diagram in Figure 4 a cluster originates from a dense region when the object's presence gets persistent, then the cluster is expanded with new points, and finally it is closed. The event that indicates the termination of the cluster expansion is the starting of a new and more recent cluster. At any instant, there is thus at most one *active* cluster and possibly one or more closed clusters. The algorithm is described in the next.

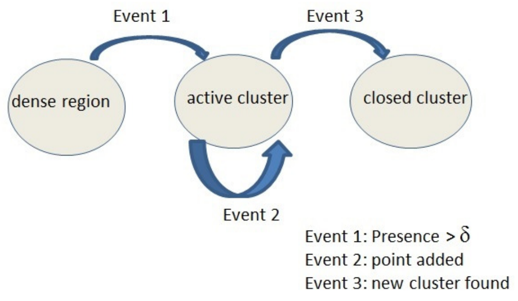


Figure 4: Cluster lifecycle

3.1 Main phases

We start by describing the main operations, next we refine the implementation aspects. Every cluster is created and next expanded until it is closed. The period in between the activation of one cluster, say C, and the activation of the subsequent cluster is called *time context* of C. The notion of time context is introduced to restrict the portion of the input trajectory contributing to the generation of the cluster, namely only the points temporally contained in the time context, can be added to the active cluster. The time context for the active cluster starts when the previous cluster is closed (or at the beginning of the trajectory) and is updated every time a new point is read.

Prior to detailing the Algorithm 1, we introduce two basic functions:

findCluster(S): the function returns the first cluster in the input trajectory S (i.e. the first in time order). If none is found, the functions returns the empty-set.

expand(activeCluster,timeContext,q): the function attempts to add the point q to the active cluster, given the time context. If successful, it returns True, False otherwise.

Initially the function *findCluster()* is repeatedly called over sub-sequences of incremental length, e.g. $T[1, i], T[1, i+i]$.., until a cluster is possibly found at time t_c. Such a cluster, say C, thus becomes active.The cluster has a start point and an end point. The time context of the cluster is initialized to $[t_1, t_c]$. During the phase of *cluster expansion*, the algorithm continues scanning the trajectory until a new cluster is found or the trajectory is terminated. At each step, the algorithm tries to append the current point p_c to the active cluster through the *expand()* function. There are two cases: (i) If the point can be added, the active cluster C grows. The end time of the cluster is the time-stamp of the current point. The scan proceeds. (ii) If the point cannot be added to the active cluster, the algorithm determines whether a cluster exists in the sub-sequence starting immediately after the end of the active cluster and the time of the current point, i.e. $[t_{end+1}, t_c]$. If it is so, such cluster becomes the active one while C is added to the sequence of *closed clusters*. The points falling in the time context but not belonging to the closed cluster are added to noise. Finally the time context is set to $[t_{end+1}, t_c]$. This guarantees that the new cluster is temporally disjoint from the previous one, i.e. C. The cycle thus repeats with the expansion of the new active cluster.

Algorithm 1 Stay Regions Discovery: SeqScan

```
procedure SeqScan(
      In: $T = [p_1, .., p_n], \epsilon, K, \delta$;
      Out: stayRegions, noise)
    $c \leftarrow 1$                                    ▷ Index scan
    $start \leftarrow 1, end \leftarrow 0$
    $activeCluster \leftarrow \emptyset$
    $stayRegions \leftarrow \{\emptyset\}$
    while $c \leq n$ do
        $timeContext \leftarrow [t_{start}, t_c]$
        if expand($activeCluster, timeContext, p_c$) then
            $end \leftarrow c$
        else
            $nextCluster \leftarrow$ findCluster($T[t_{end+1}, t_c]$)
            if $nextCluster \neq \emptyset$ then
                $stayRegions \leftarrow add(activeCluster)$
                $noise \leftarrow add(T[t_{start}, t_{end}] \setminus activeCluster)$
                $start \leftarrow end+1, end \leftarrow c$
                $activeCluster \leftarrow nextCluster$
            end if
        end if
        $c \leftarrow c+1$
    end while
end procedure
```

The final set of close clusters are the stay regions being discovered consisting of a temporal ordered sequence of sub-trajectories.

3.2 The algorithm in detail

We have shown that the resulting stay regions are temporally disjoint. Now the question is how to compute the single clusters. A straightforward implementation of the function $findCluster(S)$ is to run the DBSCAN algorithm over the set of points in S and then check whether the presence in any of the dense regions obtained in this way is persistent. The shortcoming of this approach is that, every time the function is called with an input trajectory that differs of one or few elements from the trajectory of the previous call, the dense regions have to be re-computed again. An alternative approach is to record the status of the dense regions that are progressively found and update such a status when a new point is added. The approach is described in what follows.

We use two simple data structures called *Point Descriptor* and *Dense Region Descriptor*, respectively.

- *Point Descriptor*. When the point $p_i = (x_i, y_i, t_i) \in T$ is read, the point is assigned index i, an identifier and a descriptor. The identifier is a pointer to the actual coordinates. The descriptor is the pair: $Desc(p_i) = (Neighbors, R)$ where $Neighbors$ is the set of points (identifiers) in the ϵ-neighborhood of p_i and R the possibly empty pointer to the descriptor of the dense region the points belongs to.
- *Dense Region Descriptor*. Every time a dense region is created it is assigned the descriptor (Id, Ts) where Id is the dense region identifier (e.g. progressive number) and Ts the Time Segment, defined in the next. The points belonging to the dense region j are thus the set: $\{p_i | Desc(p_i).R.Id = j\}$.

We say that a point p: (i) is a core point if it belongs to a dense regions, i.e. $Desc(p).R \neq Null$; (ii) is a border point if it is not a core point and exists a core point q with a neighborhood containing p, i.e. $Desc(p).R.Id = null \wedge \exists q, p \in Desc(q).Neighbors$. For the sake of readability, we write $N_\epsilon(q)$ to indicate *Desc(q).Neighbors*.

We recall that the Time Segment of a dense region is defined by the set of intervals in which the object is present in the region. (i.e. the gaps are omitted). It is constructed as follows. The points of a dense region are qualified as *entrances* and *exits* where a point p_i: (i) is an entrance if the preceding point in the input trajectory T does not belong to the dense region. (ii) p_i is an exit if the subsequent point in T does not belong to the dense region. A point can be both an entrance and an exit. Every pair of subsequent nodes (entrance, exit) specifies one period of presence in S. If entrance and exist coincides the period is an instant.

The whole presence in S is computed by summing up the duration of each period in the Time Segment.

3.2.1 Updating the dense regions

Let us denote with DR the set of (point and dense region) descriptors at a certain time. Detecting whether one of these regions is a cluster is straightforward: it is sufficient to consider the Time Segment specified in each of the dense regions descriptors. More complex is the expansion phase which attempts to add a point q to the active cluster. This operation is performed by the function *expand()* in the Algorithm 2.

The function creates a new point descriptor for q, while the neighborhoods of the points falling in the time context are updated (line 4); next the dense regions descriptors are updated (lines 5-7). In particular the last operation is as follows: for every point p in the neighborhood of q (i.e. $p \in N_\epsilon(q)$): (i) if q is directly density reachable from p (i.e. p is a core point in a dense region R) then q is added to R, i.e. the Time Segment of R, is updated (*LinkCorePoint(p,q)*). Note that the neighborhood $N_\epsilon(q)$ consists exclusively of points falling in the specified time context.(ii) If p has become a core point, after the addition of q, but no dense region can contain it, then a new dense region (descriptor) is created. Conversely if p is a core point but its neighbor points belong to different dense regions, these regions are merged into a unique one. Finally the Time Segment is updated accordingly (*LinkNeighbors(p)*).

It can be shown that the *expand()* preserves the properties of density-connectivity and maximality of dense regions (we omit the demonstration for lack of space). An example illustrating how the function works is reported in the next.

EXAMPLE 3.1 (UPDATING OPERATION). *Consider a trajectory of 10 points. The projection of the points on the plane is shown in Figure 5(a). Points are numbered from 1 to 10 based on the time order. The density parameters are ϵ and $N = 4$. The points are read in sequence from point 1:*

- *Point 1-4 are read. None of them is a core point as can be seen from the neighborhoods in Table 1 (first four lines):*
- *The first dense region is created at step 5 (see Figure 6(a)). After reading point 5, point 1 becomes a* core point *(the neighborhood includes 3 more points) while points 2,3,5 are* border points. *Because no other dense region contains this core point, a new dense region descriptor is created named dr_1. The corresponding Time Segment is set to: $ts_1 = [1,3] \cup [5,5]$*
- *The dense region described by ds_1 is expanded (Figure 6(b). Point 6 is read. $N_\epsilon(6) = \{1,5\}$. Point 5 becomes a core point (directly connected to points 1,2,6). Hence point 6 is added to dr_1 The time segment of dr_1 is updated to: $ts_1 = [1,3] \cup [5,6]$*
- *A new dense region is created (Figure 6(b)). The points from 7 to 9 are read. We obtain: $N_\epsilon(7) = \{3\}$, $N_\epsilon(8) = \{7\}$, $N_\epsilon(9) = \{7,8\}$. At this point, 7 is a* core point *that however is not connected to dr_1, while 3,8,9 are* border points. *A new dense region dr_2 is created with Time Segment: $ts_2 = [3,3] \cup [7,9]$.*
- *Finally the point 10 is read (Figure 6(c)). Since $N_\epsilon(10) = \{3,7,8\}$, point 3 results to be a shared* core point *between dr_1 and dr_2. The two regions are merged. As a*

Algorithm 2 Function expand()

```
 1: function EXPAND($activeCluster, timeContext, q$)
 2:     Global variables : $DR, T$
 3:     $c \leftarrow 1$
 4:     $createPointDesc(q)$
 5:     for all $p \in N_\epsilon(q)$ do        ▷ $N_\epsilon(q) \in T[timeContext]$
 6:         $linkCorePoint(p, q)$
 7:         $linkNeighbors(p)$
 8:     end for
 9:     $return(q \in activeCluster)$
10: end function
11:
12: procedure LINKCOREPOINT($p, q$)
13:     if $Desc(p).R \neq null$ then        ▷ p is a core point
14:         $Desc(q).R \leftarrow Desc(p).R$
15:         $updateTimeSegment(Desc(p).R, \{q\})$
16:     end if
17: end procedure
18:
19: procedure LINKNEIGHBORS($q$)
20:     if $isCorePoint(q)$ & $Desc(q).R = null$ then
21:         if $\nexists dr \in DR$ where $q \in dr$ then
22:             $Desc(q).R = CreateNewRegionDesc()$
23:             $updateTimeSegment(Desc(q).R, N_\epsilon(q))$
24:         else
25:             for all $dr_1, dr_2 \in DR$ where $q \in dr_1, dr_2$ do
26:                 $merge(dr_1, dr_2)$
27:             end for
28:             $updateTimeSegment(Desc(q).R, N_\epsilon(q) \setminus q)$
29:         end if
30:     end if
31: end procedure
```

$N_\epsilon(1) = \{1\}$
$N_\epsilon(2) = \{1,2\}, N_\epsilon(1) = \{1,2\}$
$N_\epsilon(3) = \{1,3\}, N_\epsilon(1) = \{2,3,1\}$
$N_\epsilon(4) = \{4\}$
$N_\epsilon(5) = \{1,2,5\}, N_\epsilon(1) = \{2,3,5,1\}, N_\epsilon(2) = \{1,5,2\}$
$N_\epsilon(6) = \{1,6,5\}, N_\epsilon(1) = \{2,3,5,1,6\}, N_\epsilon(5) = \{1,5,2,6\}$

Table 1: Neighborhoods of the first 6 points

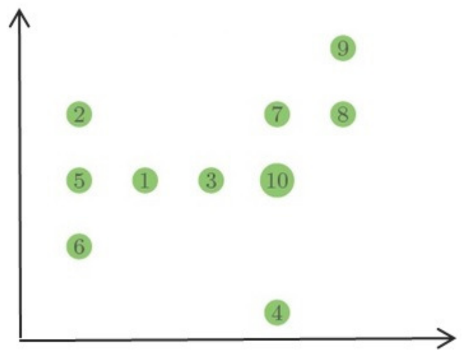


Figure 5: Set of 10 points projected on space

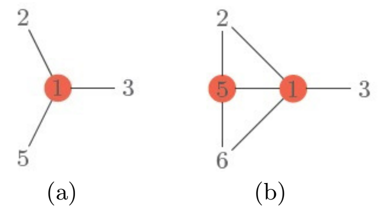


Figure 6: (a) Point 1 is a core point. A new dense region descriptor dr_1 is created. (b) Point 5 becomes a core point after point 6 is added

result also points 8, 10 become a core point. The time segment of the result is set to: $ts_1 \cup ts_2 = [1,3] \cup [5,10]$.

It can be shown that the time complexity of the algorithm is quadratic with respect to the length (number of points) of the trajectory. The time complexity is measured with respect to the costly operation computing the distance between the current point and the points in the time context. In the worst case, the input point p_i is confronted with all the preceding points $p_1, .., p_{i-1}$. The total number of distance operations is thus $\sum_{i \in [1,n-1]} i = \frac{n(n-1)}{2}$, i.e. the time complexity is $O(n^2)$, that is the complexity of the DBSCAN algorithm, if no indexing mechanism is used [23]).

4. EXPERIMENTS

We now apply and evaluate the SeqScan algorithm on the case study. Finally, we report a brief performance study, carried out using the real dataset extended with synthetic data.

4.1 Case study application

We apply the algorithm to extract the migration behavior of a group of 25 roe deer tracked in the period 2005-2008. The dataset is provided by the research institute Fondazione E.Mach. The total number of samples amounts to over 50000 points. The animals live in an area of about 30 Km^2 on the Alps near the city of Trento (Italy). The history of these animals, since their capture for the installation of the GPS collar, is known [10] and used as ground truth. The basic statistics of the dataset are reported in Table 2.

The points are sampled approximately every 4 hours, to reduce battery consumption. Under normal conditions, the distances covered in that time period are of a few hundreds

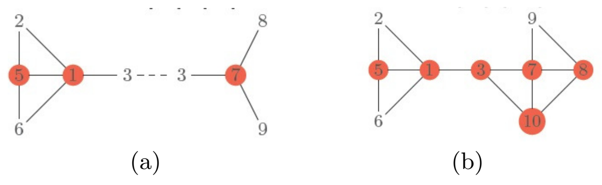


Figure 7: (a) A new dense region dr_2 is created; (b) the two regions dr_1 and dr_2 are merged because sharing a core point

of meters thus the movement is generally limited (in relation to the phenomenon of migration). A significant number of samples in the dataset are missing, due to the fact that the satellite signal is lost, for example, when the animals stay inside a dense forest, therefore the temporal distance between subsequent points varies. Trajectories are of different length and duration as shown by the high standard deviation value(σ).

Trajectories	*Avg*	σ
Δt	5.17 hours	0.93 hours
Δs	165 meters	47 meters
length	1869 points	503 points
duration	401 days	154 days

Table 2: Basic dataset statistics on: average spatial distance between two consecutive points in every trajectory (Δs), average temporal distance between two consecutive points in every trajectory (Δt), length and duration of trajectories.

4.1.1 Methodology

The goal is to identify whether the animals are either stationary or migrate and in the latter case, to differentiate the regions in which the animals stay based on presence. Stay regions are classified as large and small stay regions, depending on whether the animal is present for a long or short time. With little abuse of terminology we use the term home-range to denote a large region. Whether a large region is a home-range or not, depends on the actual definition of home-range, that is outside the scope of this analysis.

The algorithm is applied at two different stages. In the first stage, the algorithm is used to identify the large regions and runs over the entire trajectory. This phase is called *coarse analysis*. In the second phase, the algorithm is run over the sub-trajectories representing *filtered noise* i.e. long gaps and transitions, to detect the small regions. This analysis is called *fine-grained*. The values of the clustering parameters are reported in Table 3. Probably facilitated by the low number of parameters and the small group of animals, it has been possible to set a common set of values for all of the trajectories. The values are obtained empirically, on the basis of domain knowledge and data statistics. We will come back on this aspect later on.

	ϵ	K	δ
coarse analysis	100 meters	20 objects	25 days
fine-grained analysis	40 meters	5 objects	6 days

Table 3: Clustering parameters value

The animal behavior extracted from the algorithm is described both in textual and map form. The textual form follows the syntax of symbolic trajectories [24]. Specifically, every stay region is given a label l and a time period I. The label l consists of two short texts, the former indicates the type of the stay region (e.g. H stands for home-range, S for stopover and E for excursion), and the latter the identifier (i.e. $H0, H1...$). The time period I is the period of presence in the stay region denoted by the label. The symbolic trajectory for each animal is thus obtained by concatenating the descriptions: $\{I_1\ label_1\}\ \{I_2,\ label_2\}....$ The stay regions are displayed on maps as set of points enclosed in convex hulls (noise is omitted).

4.1.2 *Evaluation and discussion*

We show a few representative examples of migration behavior. The purpose is to highlight the diversity of behaviors and how such a diversity is captured by the algorithm. In particular we consider three animals, nick-named Michela, Alessandra, Lara. The sets of sampled points for each of the animals are reported in Figure 8.

```
{ ["2005-12-10 00:01:00.023"  "2006-04-14 08:01:00.023" ]"H0"}
{ ["2006-04-14 12:01:00.053"  "2007-08-04 00:02:00.011" ]"H1"}
{ ["2007-08-04 08:02:00.018"  "2007-08-18 20:01:00.049" ]"E0"}
{ ["2007-08-19 00:01:00.041"  "2008-01-07 12:01:00.051" ]"H1"}
{ ["2008-01-08 20:02:00.023"  "2008-02-29 16:00:00.054" ]"H2"}
```

(a)

```
{ ["2008-01-09 04:02:00.024"  "2008-01-09 08:01:00.029" ]"H2"}
{ ["2008-01-09 20:00:00.054"  "2008-01-10 12:02:00.006" ]"H2"}
{ ["2008-01-12 12:02:00.009"  "2008-01-12 16:01:00.047" ]"H2"}
{ ["2008-01-14 08:00:00.054"  "2008-01-14 12:01:00.047" ]"H2"}
{ ["2008-01-20 08:00:00.054"  "2008-01-20 16:01:00.027" ]"H2"}
{ ["2008-01-23 08:01:00.012"  "2008-01-24 12:00:00.053" ]"H2"}
{ ["2008-01-25 00:01:00.022"  "2008-02-02 12:01:00.024" ]"H2"}
{ ["2008-02-03 00:02:00.015"  "2008-02-21 08:01:00.023" ]"H2"}
{ ["2008-02-22 16:01:00.046"  "2008-02-23 04:01:00.042" ]"H2"}
{ ["2008-02-24 00:02:00.054"  "2008-02-25 04:01:00.042" ]"H2"}
{ ["2008-02-26 16:01:00.024"  "2008-02-26 16:01:00.024" ]"H2"}
{ ["2008-02-27 08:02:00.059"  "2008-02-28 16:01:00.051" ]"H2"}
{ ["2008-02-29 08:01:00.055"  "2008-02-29 16:00:00.054" ]"H2"}
```

(b)

Figure 10: Symbolic trajectories: (a) Michela's behavior in synthetic form, i.e. short temporal gaps are omitted. (b) The Time Segment of one of the stay regions.

- The migration behavior of *Michela* is illustrated in Figures 9(a-b). The trajectory has about 4606 points over two years. The animal moves from the home-range $H0$ to the home-range $H1$ where it makes an excursion $E0$, then moves to home-range $H2$. The symbolic trajectory at two different levels of detail (synthetic and detailed) is reported in Figure 10 (for lack of space the symbolic trajectory is only shown for this animal).

  The percentage of noise for Michela is 9.64%. The low percentage indicates that the points are highly aggregated. This measure is in line with the fact that the four stay regions are very close to each other therefore the transitions do not generate significant noise. It can also be seen that the temporal extent of the regions is aligned with the typical seasonal behavior.

- Alessandra exhibits a different behavior as illustrated in Figure 9 (c). The trajectory consists of 2642 points. In this case the animal covers longer distances to move from one region to another and that explains the high percentage of noise, i.e. 32%.

- The behavior of *Lara* is displayed in Figure 9(d). The trajectory has 1543 points, thus the animal is observed for a shorter time. During this period, the animal is stationary. This behavior reflects itself in the low percentage of noise, 1.4%.

From the experiments it turns out that 10 animals out of 25 are stationary while the others migrate. While this matches previous results in [10], the algorithm helps to thoroughly describing and finding evidence of migratory behavior in uncertain cases (e.g., the animal Michela). An interesting aspect is that, as seen before, the percentage of noise varies significantly depending on whether the animal migrates or not, as reported in Table 4. This is relevant for two reasons: first, it shows that the algorithm is able to capture the migration behavior diversity; second, it shows that noise can be a good indicator of the animals' migratory attitude. Ideally, this can facilitate large scale analysis over larger groups of animals. We recall, however, that the animals' behaviors can be contrasted (based on noise) because of the choice of a common set of values for the clustering parameters. Whether a common set of parameter values can be found with larger groups of animals will be investigated as part of future work.

Group of animals	#	avg(%noise)	σ
stationary	10	2.8	1.9
migrators	15	20	9.9

Table 4: Average and deviation standard of the noise distribution for the groups of stationary and migrating animals

In a few cases we have noticed an interesting divergence between the outcome of the algorithm and visual analysis. The specific situation is sketched in Figure 11: an animal stays in the region A then moves to reach region B. Progressively however the region B expands until it overlaps A. In this case the algorithm considers A and B two different clusters, thus if the clusters are large regions, it means that the animal migrates. Conversely the visual analysis leads to consider this set of points as a unique region.

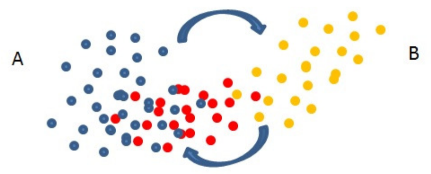


Figure 11: Cluster A: blue points; cluster B: yellow points in B. The red points highlight the expansion of B towards A

4.2 Performance evaluation

The above dataset is used for the run time performance study. The goal is to demonstrate experimentally that the time complexity is quadratic with respect to the number of points of the input trajectories. We recall that no indexing mechanism is used to speed up the operation [23]. The experimental setting is as follows. The algorithm is written in Java (Jdk 1.7)). It runs on Sony Vaio with Cpu i7-3632Qm with 2.20GHz and 8 GB RAM. For the experiment,

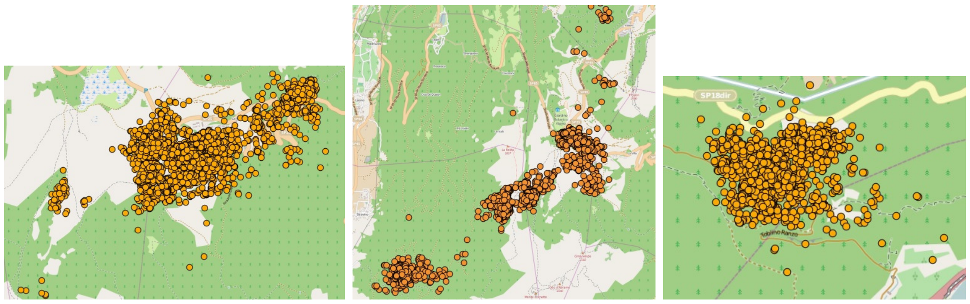
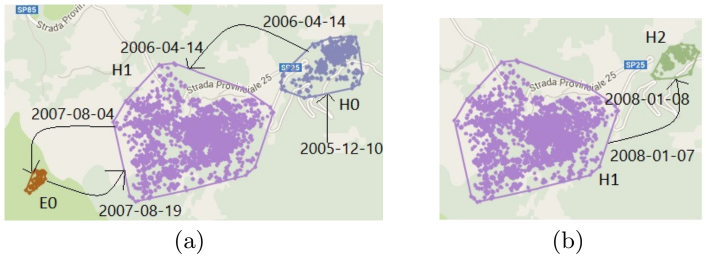


Figure 8: Sampled points for the example animals *Michela, Alessandra, Lara* (QGIS maps)



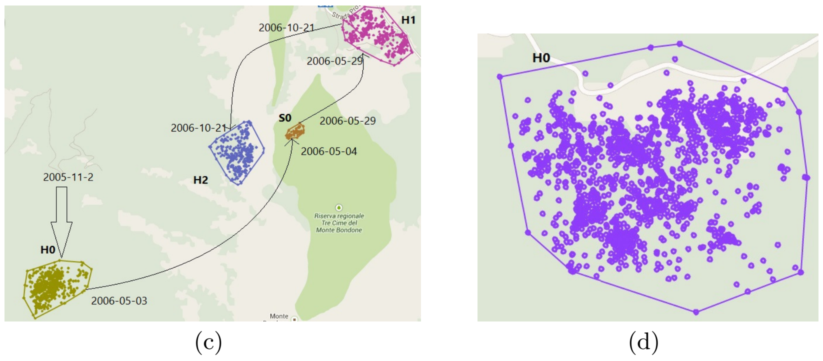


Figure 9: (a-b)*Migration pattern of Michela*: from H0 to H1; and from H1 to H2. (c)*Alessandra*: the animal covers longer distances. (d)*Lara*: stationary animal

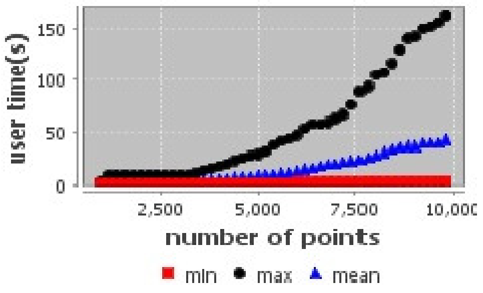


Figure 12: Run-time performance over the dataset

we use the previous trajectory data set, extended with synthetic data. The purpose of synthetic data is to allow the generation of trajectories of arbitrary length. In particular, the additional points are simply obtained by modifying the time-stamp, i.e. $t_m = t_{m-1} + 4h + random$ where $random \in [-2h, 2h]$. We consider the value of n ranging from 1000 to 10000 points for the 25 trajectories. For each value of n, the minimum, maximum and average run time is computed for the 25 trajectories. The resulting graph that confirms the complexity analysis is shown in Figure 12.

5. RELATED WORK

The SeqScan algorithm relates to diverse density-based clustering models. A major one is incremental clustering. In particular, our work takes inspiration from IncrementalDBSCAN [19], a solution drawn to update existing DBSCAN clusters. We apply a similar approach, namely the dense regions are progressively updated, however the updating strategy is different. ST-DBSCAN is an extension of DBSCAN for the clustering of events located in space and time [25]. In addition to the spatial neighborhood radius, ST-DBSCAN considers a temporal neighborhood radius. Thus, a point is considered as core when the number of points in the neighborhood is greater or equal to the threshold value within spatial and temporal thresholds. By contrast, in SeqScan the neighborhood has exclusively a spatial meaning because any assumption on how space and time are related would be arbitrary. Temporal thresholds are also used for the detection of stops in high-sampling-rate trajectories [13, 22]. The temporal threshold in [13] specifies the minimum duration of the stop (we recall that duration is different from the concept of presence). The technique, however, does not provide guarantees that subsequent stops are temporally disjoint. This pitfall is avoided in [22]. However, also in this case the neighborhood consists of points that are close in space and time, while we recall that the temporal closeness of points is not required in our model. A common deficiency of these methods is the lack of validation on real applications.

6. CONCLUSION

The paper presents a novel framework for the study of migratory behaviors, consisting of: a mobility pattern model; an algorithm, i.e. SeqScan, to extract such a pattern from GPS trajectories; a usage methodology for the study of wild animal migrations; and a first validation of the whole framework over a real data set. On the algorithmic side, SeqScan provides a conceptually clean and founded mechanism for the analysis of stop-and-move like patterns over large temporal scales. Further properties, such as scalability issues, will be analyzed in more detail in the near future. On the application side, we plan to extend the use of the proposed approach in the context of animal movement, e.g. to several species/populations. This is especially valuable in the light of recent considerations on the plasticity of animal behavior (the *stationarity-migratory continuum*). The use of the percentage of noise as a quantitative index of "migratoriness" is especially interesting and innovative, offering the opportunity to fine-tune the space-time granularity at which analyzing movement patterns. This also opens up the study of human mobility over large temporal scales.

7. REFERENCES

[1] 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):42:1–42:32, 2013.

[2] J. Gudmundsson, P. Laube, and T. Wolle. Movement patterns in spatio-temporal data. In *Encyclopedia of GIS*, pages 726–732. Springer US, 2008.

[3] Y. Zheng, L. Zhang, X. Xie, and W. Ma. Mining interesting locations and travel sequences from gps trajectories. In *Proc. WWW*, 2009.

[4] S. Dodge, R. Weibel, and A-K Lautenschütz. Towards a taxonomy of movement patterns. *Information Visualization*, 7(3-4):240–252, 2008.

[5] F. Cagnacci, L. Boitani, R. Powell, and M. Boyce. Animal ecology meets gps-based radio telemetry: a perfect storm of opportunities and challenges. *Phil. Trans. R. Soc. B*, 365:2157–2162, 2010.

[6] B. Chapman, C. Brönmark, J-A. Nilsson, and L.A. Hansson. The ecology and evolution of partial migration. *OIKOS*, (120)12:1764–1775, 2011.

[7] S. Spaccapietra, C. Parent, M.L. Damiani, J.A. de Macedo, F. Porto, and C. Vangenot. A conceptual view on trajectories. *Data Knowl. Eng.*, 65(1):126–146, 2008.

[8] W. H. Burt. Territoriality and home range concepts as applied to mammals. *Journal of Mammalogy*, 24:346–352, 1943.

[9] J.G. Kie, J. Matthiopoulos, J. Fieberg, R.A. Powell, F. Cagnacci, M.S. Mitchell, J-M. Gaillard, and P.R. Moorcroft. The home-range concept: are traditional estimators still relevant with modern telemetry technology? *Philos Trans R Soc Lond B Biol Sci.*, 365:2221–2231, 2010.

[10] F. Cagnacci, S. Focardi, M. Heurich, and al. Partial migration in roe deer: migratory and resident tactics are end points of a behavioural gradient determined by ecological factors. *OIKOS*, 120 -12:1790–1802, 2011.

[11] X. Cao, G. Cong, and C. S. Jensen. Mining significant semantic locations from gps data. *Proc. VLDB Endow.*, 3(1-2):1009–1020, 2010.

[12] M. Buchin, A. Driemel, M. van Kreveld, and V. Sacrist. Segmenting trajectories: A framework and algorithms using spatiotemporal criteria. *Journal of Spatial Information Science*, 3:630–660, 2011.

[13] Z. Yan and D. Chakraborty. *Semantics in Mobile Sensing.* MorganClaypool, 2014.

[14] L.O. Alvares, V. Bogorny, B. Kuijpers, J. de Macedo, B. Moelans, and A. Vaisman. A model for enriching trajectories with semantic geographical information. In *Proc. ACM GIS*, 2007.

[15] H. Yoon and C. Shahabi. Robust time-referenced segmentation of moving object trajectories. In *Proc. ICDM*, 2008.

[16] A. Anagnostopoulos, M. Vlachos, M. Hadjieleftheriou, E. Keogh, and P. Yu. Global distance-based segmentation of trajectories. In *Proc. of ACM SIGKDD*, 2006.

[17] M. Buchin, H. Kruckenberg, and A.Kölzsch. Segmenting trajectories based on movement states. In *Proc. SDH*, 2012.

[18] M. Ester, H.P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD*, 1996.

[19] M. Ester, H.P. Kriegel, J. Sander, M. Wimmer, and X. Xu. Incremental clustering for mining in a data warehousing environment. In *Proc. of VLDB*, 1998.

[20] F. Cao, M. Ester, W. Qian, and A. Zhou. Density-based clustering over an evolving data stream with noise. In *SDM*, 2006.

[21] M. Ankerst, M. Breunig, H.P. Kriegel, and J. Sander. Optics: Ordering points to identify the clustering structure. In *Proc. of the 1999 ACM SIGMOD*, 1999.

[22] M. Zimmermann, T. Kirste, and M. Spiliopoulou. Finding stops in error-prone trajectories of moving objects with time-based clustering. In *Intelligent Interactive Assistance and Mobile Multimedia Computing*, volume 53, pages 275–286. 2009.

[23] J. Sander, M. Ester, H.P. Kriegel, and X. Xu. Density-Based Clustering in Spatial Databases: The Algorithm GDBSCAN and Its Applications. *Data Mining and Knowledge Discovery*, 2(2):169–194, 1998.

[24] F. Valdés, M.L. Damiani, and R.H. Güting. Symbolic Trajectories in Secondo: Pattern Matching and Rewriting. In *DASFAA*, 2013.

[25] D. Birant and A. Kut. ST-DBSCAN: An Algorithm for Clustering Spatial-temporal Data. *Data Knowl. Eng.*, 60(1):208–221, 2007.