Finding Stops in Error-Prone Trajectories of Moving Objects with Time-Based Clustering

Max Zimmermann[1], Thomas Kirste[2], and Myra Spiliopoulou[1]

[1] Otto-von-Guericke-University of Magdeburg, Germany
{max.zimmermann,myra}@iti.cs.uni-magdeburg.de
[2] Dept. Computer Science, Rostock University, Germany
thomas.kirste@uni-rostock.de

Abstract. An important problem in the study of moving objects is the identification of *stops*. This problem becomes more difficult due to error-prone recording devices. We propose a method that discovers *stops* in a trajectory that contains artifacts, namely movements that did not actually take place but correspond to recording errors. Our method is an interactive density-based clustering algorithm, for which we define density on the basis of both the spatial and the temporal properties of a trajectory. The interactive setting allows the user to tune the algorithm and to study the stability of the anticipated *stops*.

1 Introduction

There is an increasing number of applications on moving objects e.g. satellites or RFID that require reliable tools to discover patterns and analyze trajectory structure. Recent technological improvements allow capturing and recording huge data volumes on moving objects. Each moving object is represented by a large number of space-time points. All points connected along time give a trajectory which represents the path of an object. Thus, an efficient method for mining trajectories is necessary to get knowledge out of such data.

We are particularly interested in the analysis of trajectories that reflect human behavior, such as the activities of a person who suffers from a chronic disease, the route of a truck transporting products, the movements of RFID-tagged clothes in a department store etc. There are two problems that make trajectory mining in those applications more difficult. First, human activity is not predictable not even for the types of activity as assumed e.g. in the hierarchical activity models of [LFK07]; the movement of a person attempting to steal an RFID-tagged apparel in a store is such an example. Furthermore activity is influenced by external factors, expressed e.g. in the form of unexpected transportation modes. The data recording may be partially unreliable, for example when persons move through areas that distort signals.

For the first problem, consider the trajectory of a person who spent some time walking and some time driving. The search for *stops* should not misinterpret movement at walking pace as *stops*. Moreover, assumptions about location of *stops* should be avoided, since the person may abandon predefined pathways.

D. Tavangarian et al. (Eds.): IMC 2009, CCIS 53, pp. 275–286, 2009.

The second problem may occur when receiving signals from satellites, e.g. in GPS devices. Current GPS technology may not work reliably inside buildings or in areas where sky view is obscured by trees [KLK+02]. Various dysfunctions may appear such as transcription errors or misleading recordings. In our experiments, we use a trajectory that contains such artifacts, namely recordings of motion when the GPS device was indoor and not moving.

Our paper presents a clustering and visualization approach for the discovery of *stops* in a trajectory, including *stops* masked as motion due to artifacts. Our method is based on OPTICS [ABKS99], a density-based interactive clustering algorithm that we extend to capture temporal proximity of data points on the basis of elapsed time and speed of motion. Our new method further incorporates a visualization utility that depicts both the motion and the duration of a *stop*.

The rest of the paper is organized as follows. In the next section we describe related work on mining trajectories. Section 3 describes our spatio-temporal trajectory model. In section 4 we present the methods for discovering and visualizing *stops* on the basis of this model. We describe our experiments on a trajectory containing *stops* and motion artifacts in section 5. The last section concludes our study and provides an outlook.

2 Related Work

In recent years, there is a proliferation of research advances on trajectory mining. A typical mining task is to discover crowded areas like airports [PBKA08]. We distinguish between two categories of methods: those that analyze a single trajectory [PBKA08] and those that derive patterns from multiple trajectories [NP06]. Our work belongs to the first category.

The algorithm SMoT which has been described by [ABK+07], divides a trajectory into *stops* and *moves*. It needs, however, a priori information for finding *stops* which is a big disadvantage. A derivation of SMoT called CB-SMoT has been presented in [PBKA08]: it needs no a priori information but can exploit such information CB-SMoT uses a speed-based method for detecting clusters and associates each cluster with a *stop*. For clustering it uses a variation of DBSCAN [EKSX96], a well known density-based clustering algorithm that even discovers unrecorded *stops*. However, it has difficulties in discovering *stops* after motions at different speeds, e.g. different transportation modes, since it relies on a global parameter that defines the size of the considered space to compute the speed in a certain area. Figure 1 depicts a counterexample: there is much variation of speeds; the speed changes at A-D are *stops*. The *stops* B and D would not be discovered by CB-SMoT if the speed parameter value were set below B and D. Thus, the accuracy of CB-SMoT is affected by the value of the speed parameter.

DBSCAN is a density-based clustering algorithm, especially designed to be robust towards noise [EKSX96]. It has a local notion of proximity and is sensitive to the order of reading the data points. These properties make it particularly appealing for trajectory mining, since the recordings of motion are ordered. Indeed, DBSCAN and its interactive follow up OPTICS [ABKS99] are often used for trajectory mining [Han05], [ZLC+08].

Our variation of OPTICS works with the model proposed in [SPD+08]: Spaccapietra et al. capture *stops* and *moves* in a trajectory, whereby a *move* captures the semantics of fast movements such as acceleration, driving a car, while a *stop* reflects slow motion as in a traffic jam, complete absence of motion as stopping at a traffic light or being indoor. We extend their model by incorporating temporal information in the definition of density. Differently from CB-SMoT, our method uses an duration threshold as a parameter instead of a global parameter to find *stops* and *moves* of a trajectory.

By using this threshold it can find *stops* associated with different speeds. In figure 1 it is sufficient to set the threshold at the height of B to discover all *stops*. Additionally the results of our method can be used to explore unpredictable behavior or technical dysfunctions.

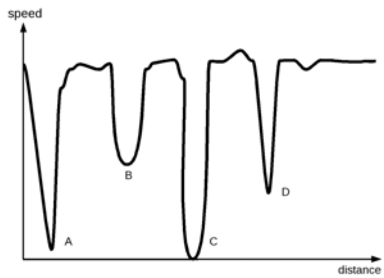


Fig. 1. Stops of different speeds

3 Spatio-Temporal Modeling of Trajectories

In the following, we describe the model of trajectory. We then describe the properties of OPTICS. In subsection 3.3 we introduce our notion of time-based neighborhood, on which our new version of temporal OPTICS is based.

3.1 Stops and Moves on a Trajectory

In this section we give a definition of a trajectory and the semantic properties of the model from [SPD+08]. Both of them takes an important role of our approach and will be used in the remaining of the paper.

A trajectory consists of *stops* and *moves*. A *stop* is a sub-trajectory confined to a certain area and characterized by very low speed.

Definition 1. *(trajectory)*

Let P *be a list of space-time points,* p_i *be a point at* (x_i, y_i, t_i) *where* $x_i, y_i \in \Re, t_i \in \Re^+$*. A trajectory is an ordered* P *for* $i = 0, 1, ..., N,$ *and* $t_0 < t_1 < ... < t_N$*.*

To compute the speed it requires a geometrical and a time aspect. For the geometrical aspect we use a maximal length parameter that defines the allowed distance between two consecutive points to be in a *stop*.

The time aspect is captured by considering a *stop* as a set of points that are recorded within a time interval. Suppose a car driver has to slow down because of a red traffic light and the minimal speed has been defined as 10 km/h. Then, a *stop* starts if the driver reaches a less than 10 km/h and ends when the driver accelerates over 10 km/h.

Definition 2. *(stop)*
Let T be the trajectory, S be a sub-trajectory of T for the the time interval $[t_i, t_{i+n}]$ with temporal disjoint points and $\lambda_{distance}$ be the minimal length of the stop. *Then, S is a* stop *if $\{p_i = (x_i, y_i, t_i), p_{i+1} = (x_{i+1}, y_{i+1}, t_{i+1}), ..., p_{i+n} = (x_{i+n}, y_{i+n}, t_{i+n})\}$, where $distance(p_j, p_k) \leq \lambda_{distance}$ for $[i, i+1, \ldots, j, j+1, \ldots, k]$ with $j \neq k$.*

If a point does not belong to a *stop*, it is part of a *move*.

Definition 3. *(move)*
Let S_i and S_j be two stops *of a trajectory T with the length of n and m, respectively, with $i + n < j$. Then, a* move *can appear in the following intervals, $[t_k, t_{i-1}]$, $[t_{i+n+1}, t_{j-1}]$ or $[t_{j+m+1}, t_l]$ for $k \geq 0$ and $l > (j + m + 1)$.*

Note: each *move* can become a *stop* if the minimal speed is set to a larger value. To distinguish between *moves* and *stops* we propose a time-oriented version of OPTICS. The next subsection will give a short introduction to OPTICS.

3.2 Core Points and Density in OPTICS

OPTICS is a interactive density-based clustering method [ABKS99]. The basic idea of density-based clustering is that each object of a cluster must have at least *minPts* many points within a given radius *eps* (neighborhood). It has been shown [ABKS99] that density-based clustering discovers clusters of arbitrary shape in noisy areas. As human behavior has no explicit structure the noise tolerance of arbitrary shapes is a unavoidable feature.

The output of OPTICS is a cluster-ordering; the left upper picture of Table 1 depicts the cluster-ordering for $eps = 100$ meters. To order the points to a list, the OPTICS describes a point only by two values: the *core-distance* and the *reachability-distance*. Both of them depend on [EKSX96] definition of a *core point* and the ε-neighborhood, introduced in the following.

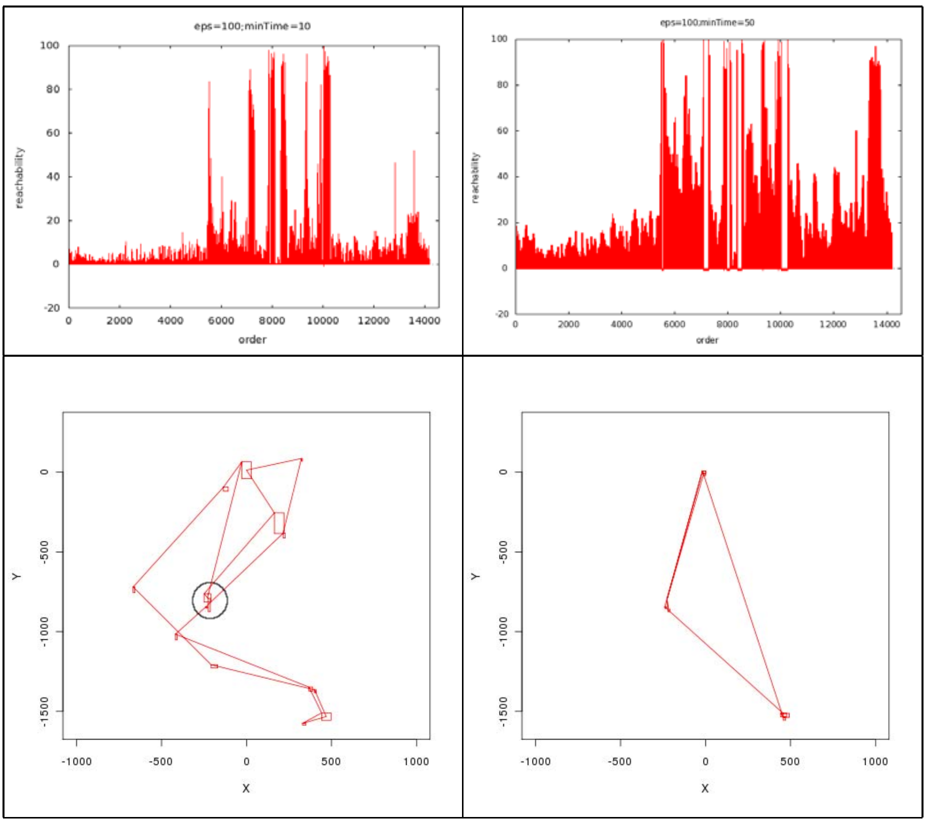
Let p_i be a point from a dataset P with the length N, $d(x, y)$ be a distance function and let ε be the threshold. The ε*-neighborhood* of point p_i is defined as $N_\varepsilon(p_i) = \bigcup_{n=0}^{N} \{p_n \mid (d(p_i, p_n) \leq \varepsilon \wedge p_i \neq p_n\}$. A point is a *core point* when its ε-neighborhood contains at least *minPts* points, i.e. $N_\varepsilon(p_i) \geq minPts$. Then, this point also belongs to a cluster. $\varepsilon/minPts$ gives the minimal dense that a region must have to be part of a cluster.

Based on the ε-neighborhood and *core point*, we define the point values for OPTICS. The *core-distance* of a point p is the distance to the *minPts* closest point so that p becomes a *core point*. The second value that OPTICS uses to describe a point is the *reachability-distance* from a point p_i to a point p_j. If p_i is further away than the *minPts* neighbor of p_j, the *reachability-distance* is the normal distance from p_i to p_j. Otherwise it is defined as the *core-distance* of p_j if p_j is a *core point*. If not, the *reachability-distance* between p_i and p_j is ∞.

OPTICS uses the *reachability-distance* as follows. Initially a random point p_0 is chosen. Then, we iterate over all unvisited points and choose at each iteration step i, the point as next that has the smallest *reachability-distance* of all unvisited *core points* with respect to p_{i-1} as point. The iteration *stops* when all objects of the data set have been considered.

The result of the OPTICS may be a 2D reachability plot with the visiting order of the points on the abscissa and the *reachability-distance* of each corresponding point on the ordinate. A high reachability value means that all the successors have a high distance to all the predecessors. Hence, areas surrounded by high values of reachability may be defined as dense areas.

Table 1. Results for two runs of T-OPTICS (minTime=10 and 50) with alpha=70



3.3 Spatio-Temporal Neighborhoods

Our OPTICS variation is supposed to distinguish between fast and slow areas. According to definition 2 a candidate *stop* describes areas of a trajectory where the object moves slowly. To define a *stop* we have to establish a relation between space and time. Hence, we also consider temporal proximity when we define neighborhoods, thus ensuring that data points that are spatially close but have been visited temporally far apart do not become part of the same neighborhood. We define this formally below and then proceed to describe our extension to OPTICS.

In a simple 2D environment, the distance between two points is their geographical distance. To compute the distance of two points p_i and p_j on a trajectory we must consider the path connecting them and must also take account of the time elapsed for going from one point to the other. The geographical distance may be computed by any known L-norm. In our examples we use the Euclidean distance $d()$ to compute distances in a 2D space.

Definition 4. *(trajectory-distance)*

Let p_i and p_j be two distinct points of a trajectory. We define their distance as $Tdistance(p_i, p_j) = \begin{cases} \sum_{n=i}^{j-1} d(p_n, p_{n+1}), \ \textit{if } \{t_i < t_j\} \\ \sum_{n=j}^{i-1} d(p_n, p_{n+1}), \ \textit{else} \end{cases}$

We use this new notion of distance when defining the term *core point*. We consider two kinds of neighborhood, one capturing only the spatial aspects of the trajectory, the other capturing also time.

Definition 5. *(trajectory neighborhood)*

Let T be a trajectory of length N, p_i a point in the trajectory and let eps denote the radius of a neighborhood around a data point p_i, similarly to the ε in the original OPTICS. Then, the trajectory neighborhood of p_i is defined as

$$TNeighbourhood(p_i, eps) = \{ \ p_n \mid (t_n > t_i \wedge Tdistance(p_i, p_n) \leq eps) \vee (t_i > t_n \wedge Tdistance(p_n, p_i) \leq eps) \ \}$$

where we ensure that the trajectory neighborhood of a data point contains data points that were reached before or after p_i.

We now restrict a trajectory neighborhood to contain only data points reached within a specific elapsed time.

Definition 6. *(core neighborhood)*

Let p_i be a point in trajectory T, eps be the geometrical aspect and minTime be the time aspect. Then, the core neighborhood *is defined as*

$$CNeighbourhood(p_i, eps, Mintime) = \{ \ p_n \mid p_n \in TNeighborhood(p_i, eps) \wedge |ti - t_n| \leq Mintime \ \}$$

Definition 7. *(core point)*

Let p_i be a point in trajectory T and TN its neighborhood. Then, p_i is a core point if $CNeighborhood(TN, minTime) \neq \emptyset$.

Contrary to OPTICS we define a *core neighborhood* by having at least one neighbor in its *core neighborhood* (Def. 6). That property ensures a slight influence by missing points. We can illustrate this by two kolmogorov axioms [Kol33].

Assume we have a set A consisting of a point p_i and one core neighbor of the p_i *core neighborhood* and a set B with all core neighbors of the p_i *core neighborhood* so that $A \subset B$. In addition to it, the probability that a point is being missed is for all points equal. Let us look for the following probabilities,

$$P(a\ point\ of\ A\ is\ missing) = P(A) \text{ and } P(\ some\ point\ is\ missing) = P(B).$$

Since A is a subset of B we can use the second and third kolomogorov axiom. This results in $P(B) = P(A) + P(B \setminus A)$ and it follows that $P(A) \leq P(B)$.

Using the new notion of neighborhood (Def. 6) that captures space and time, we next introduce our algorithm for the discovery and presentation of *stops*.

4 Stop Discovery and Visualization

We come now to describe our approach for stop discovery. We designed stop discovery as an interactive process, for which we devised *T-OPTICS*, a variation of OPTICS [ABKS99] for time-based neighborhoods, and a visualization utility.

We first specify the notion of cluster; a cluster corresponds to a *stop* on the trajectory. The interactive clustering algorithm is presented thereafter. In the last subsection we describe the visualization of the discovered *stops*.

4.1 Cluster as Stops

To use clustering for stop discovery, we (re)define the notion of reachability to capture areas of slow motion. For a data point p_i, we define the length of its *core neighborhood* (Def. 6) on the basis of the trajectory distance (Def. 4). Similarly to OPTICS, we compute the reachability values of all points and define a cluster as sequence of points among two high reachability values. These data points are temporally proximal (by Def. 6) and are spatially close (by Def. 5). Thus, they correspond to an area of slow (or no) motion - a *stop*.

Definition 8. *(trajectory-core-length)*
Let $p_i \in T$ be a data point and CN its core neighborhood. *Its trajectory-core-length is the maximum trajectory distance within CN, if it is non-empty:*
$tcorelength(p_i) = \min\{\max_{p_j \in CN} Tdistance(p_i, p_j), \infty\}$

Equally to the reachability distance of the original OPTICS, we define the *reachability-length* of a data point p_j with respect to another data point p_i.

Definition 9. *(reachability-length)*
Let p_i and p_j be points of a trajectory, such that $t_i < t_j$. The reachability-length of p_j from p_i is defined as the maximum between the trajectory core length of p_i and the trajectory distance between the two points:
$reachlength(p_i, p_j) = \max\{tcorelength(p_i), Tdistance(p_i, p_j)\}$,
implying that if p_j is in the core neighborhood *of p_i, then reachability-length of p_j w.r.t p_i is the trajectory-core-length of p_i.*

Similarly to OPTICS, a cluster is a sequence of points that are close to each other and surrounded by points that exhibit very high reachability values. A threshold α is used to specify when a reachability value is large enough to serve as a cluster's margin. More formally:

Definition 10. *(α-Cluster)*
Let $T = \{p_1, \ldots, p_N\}$ be a trajectory and let R be the ordered list of reachability values for points in T. Let $r_i, r_j \in R$ be values with $0 \leq i < j - 1$, so that $\min\{r_i, r_j\} \geq \alpha$ and $\max\{r_{i+1}, r_{i+2}, ..., r_{j-1}\} \leq \alpha$. Then, the set $\{r_{i+1}, r_{i+2}, ..., r_{j-1}\}$ constitutes a cluster to α, also denoted as α-cluster.

For example, in figure 2 we show the reachability values (y-axis) of more than 14000 data points. Each horizontal line corresponds to a certain α and results in different clusters. The lowest threshold is at 4: it produces a large number of clusters that correspond to many small areas of slow motion. For $\alpha = 14$, we get fewer clusters, some of which cover large areas; the first 5500 objects would constitute one big cluster. For $\alpha = 28$, we get a very small number of clusters.

In the next subsection we describe how the parameters for the clustering are computed, how neighborhoods are built and data points are ordered.

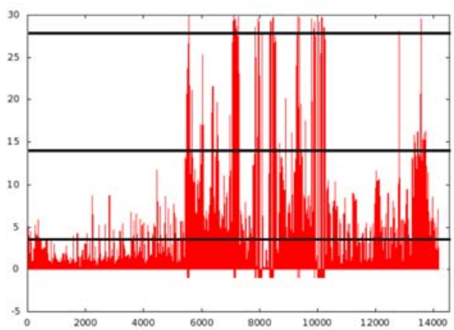


Fig. 2. Illustration of cluster constellations for different values of the threshold α

4.2 Interactive Cluster Construction

The pseudo code below depicts our algorithm `StopFinder` at a very abstract level. The algorithm consists of four steps. We explain the first three steps on stop discovery here. The visualization is described in subsection 4.3.

Algorithm 1. StopFinder

1: estimate *eps* and $minTime$
2: invoke T-OPTICS $(eps, minTime)$
3: extract potential *stops* on the ordered list for different values of α (interactively)
4: invoke visualization method

In the first step (line 1) we estimate the spatial parameter *eps* and the temporal parameter $minTime$ for the specification of neighborhoods. To this purpose, we perform a first approximation of *eps* by computing the trajectory-distance from p_1 to p_N and averaging it over the number of data points N. We similarly approximate $minTime$ by averaging the duration of the trajectory over N. We then compute the ratio of those computed values $\overline{eps}$ and $\overline{minTime}$ and use it as basis for the specification of *eps* and $minTime$.

We might use $\overline{eps}$ and $\overline{minTime}$ directly as values of $eps, minTime$. However, the computation of the ratio allows us a more elaborate tuning. For example, if the ratio $\overline{eps}$: $\overline{minTime}$ is 1 : 3, then the average time needed to go from data point p_i to the next one p_{i+1} is 3 units. Since a data point requires at least one member in its *core neighborhood* to be a *core point* (cf. Def. 6 and Def. 7), setting $minTime$ to less than 3 units (for $eps = 1$) may yield too many data points unreachable from most others. The smaller *eps*, the smaller becomes the trajectory neighborhoods and with them the likelihood that a *core neighborhood* will have one member. In contrast, for a bigger *eps* the ratio may be smaller (e.g. 100 : 50) as the likelihood of having one neighbor increases. Hence, by using the ratio, we can derive one parameter if the other is specified.

In the second step (line 2), T-OPTICS uses our new concepts for trajectory distance, *core point*, trajectory-core-length and reachability-length when building neighborhoods, identifying *core points* and computing the reachability of each data point. A crucial difference between OPTICS and T-OPTICS is in the ordering of the data points. For OPTICS, the ordering does not need to reflect the sequence of the points in the trajectory. For T-OPTICS, there is no re-ordering; the data points appear in the same order as in the trajectory, but are now associated with reachability values. These values allow the identification of the clusters (line 3), subject to the threshold α (cf. Def. 10).

Once identified, the *stops* must be mapped in terms of the original trajectory (see e.g. a trajectory in Table 2 left picture and its three *stops* as ends of a triangle in Table 1 lower right). For a α the third step computes the cluster in the way that has been described in section 4.1. The visual representation of the *stops* should encompass more than their spatial coordinates, though: the duration of each *stop* contributes much to understanding the semantics of the *stop* and the behavior of the object. The visualization mechanism (line 4) described in the next subsection allows the inspection of the spatio-temporal properties of the identified *stops*.

4.3 Visualization Method

After extracting clusters from T-OPTICS, we must visualize them in such a way that the observer associates the clusters with locations of the trajectory. We use a rectangular representation of clusters. In the following we explain this process.

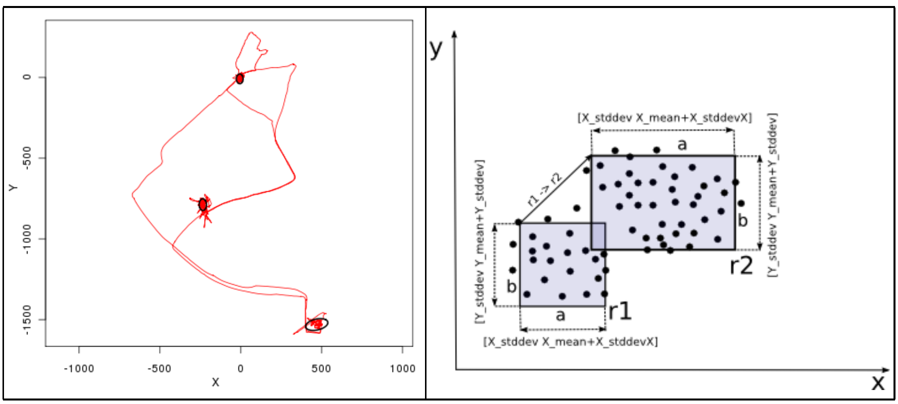
Def. 10 defines a cluster as accumulation of points that are spatially and temporally very close. To accumulate the points of a cluster, we consider each cluster as separate data set with 2 attributes: x-dimension and y-dimension. Then, for each attribute we compute mean and standard deviation. With this we obtain the four sides of a rectangle as it can be seen in Table 2 (right picture). Side b yields from the line $[Y_{stddev} Y_{mean} + Y_{stddev}]$ and side a from $[X_{stddev} X_{mean} + X_{stddev}]$.

To model the temporal aspect of a cluster, we should keep in mind that points in the same cluster are temporally very close. Hence, it is sufficient to connect each rectangle with the next one in terms of time: the result is a rough representation of the trajectory, consisting of the *stops* and lines that connect them. For example, the edge $r_1 \rightarrow r_2$ on Table 2 (right picture) indicates that cluster r_1 occurred before cluster r_2. Two rectangles may be unified if the corresponding clusters share a certain percentage of intersection, the percentage is defined by a threshold; e.g. the value 70 means that two rectangles which intersect more or equal to 70 % will be represented by a unification.

Through this we obtain on the one hand a condensed illustration of areas in the trajectory that are temporal different but spatial pretty close. This could be the daily errand to the supermarket, which can be in the morning or in the evening. Our visualization joins this to one rectangle with different edges in time. On the other hand the threshold helps us to distinguish dense and less dense clusters. Two rectangles that intersect less than the threshold are probably no daily habits of the recorded object. Rather, it allows us the interpretation of cluster, e.g. as an artifact.

In the next section we show the evaluation of our `StopFinder`.

Table 2. Left: original GPS trajectory; right: two clusters, represented as rectangles and connected across the time axis



5 Experiments

In this section we study the behavior of our `StopFinder` on a trajectory that contains artifacts. At first we describe the experimental data; then we present the chosen parameters. We present the results for the real data set at the end and show an example for the interpretation of our rectangle visualization.

The trajectory used in our experiments contains recordings of a person's movements for two consecutive days with the different transportation modes walking, cycling and driving a car. At the first day we were recording five hours and on the second day it has been two hours. The recording device was a customary

GPS device. In Table 2 (left picture) we see the trajectory. Three potential *stop* areas are visible. Among them, the motion at center is actually an artifact; there has been no motion, so this is in fact a *stop*.

The data set that we use for our experiments includes no information about *stops* or artifacts. It consists only of spatio-temporal points that describe the trajectory; every 1-4 seconds the longitude and latitude of the human position has been recorded as a spatio-temporal point. Thus, finding all *stops* and distinguish between *stops* and artifacts is the aim. Because of the advanced knowledge about the trajectory we have a ground truth for the evaluation.

For our experiments we invoke the `StopFinder` with the described data set. The estimation of parameter *eps* and $minTime$ yields to the ratio $1 : 3$. As we know roughly the variety of transportation, we bound a *stop* to a radius of 100 meters ($eps = 100$) and to a $minTime$ of 10 and 50 seconds. The upper left picture and the upper right picture of Table 1 show the reachability values computed by T-OPTICS. for In both pictures we see two valleys, one at the start and one at the end. They are clearly clusters and are extracted as such.

For extracting potential *stops* from the reachability values we set α (cf. Def. 10) to 70. The two lower pictures of Table 1 show the *stops* with our rectangle visualization. By depicting a cluster as a rectangle rather than a single point, we acquire better insights. For instance, the black surrounded area of the left lower picture of Table 1 shows two overlapped rectangles. We see that the the artifact is captured by overlapping rectangles, while the two other *stops* that contain no motion are captured by a single rectangle. In general, short edges of the rectangle indicates that there has been actually very dense movement. In this trajectory, this corresponds to the artifact motion that is actually a *stop*.

We see that our StopFinder has identified all three *stops* of the trajectory. We also see the impact of large vs small $minTime$ values: as $minTime$ decreases, we see more *stops*. Keeping in mind that a *stop* is actually a very slow motion, small $minTime$ values capture short *stops*, as for direction change. Large values of $minTime$ result in fewer *stops*, whereby the artifact (slow motion that IS a *stop*) is recognized in all cases. Through the variation of $minTime$ we can identify *stops* resulting of different transportation modes.

6 Conclusions

In this paper we proposed an interactive density-based clustering algorithm, aimed at discovering *stops* in a trajectory that contains artifacts of motion and different transportation modes. We augmented the density with the spatial and the temporal properties of an trajectory. We have further designed a visualization mechanism that depicts both the spatial and temporal properties of the *stops* identified by the clustering algorithm. The first experiments with our approach show that we can discover *stops* despite the existence of movements at different speeds and that we can even identify artifact motions.

There are several opportunities for future research. Our estimations rely on mean and standard deviation; the mean is sensitive to outliers, though. So, we intend to study more robust methods for the estimation of our parameters such as confidence intervals. Distinguishing between *stops* that persist for different values of *minTime* and those that are just very slow motions, is another subject of future work. Another ongoing work yields from adding more features to T-OPTICS, as for instance an angle change rate for regarding direction changes. Zheng et al. [ZLC+08] have shown that this is a promising option.

References

[ABK+07] Alvares, L.O., Bogorny, V., Kuijpers, B., de Macêdo, J.A.F., Moelans, B., Vaisman, A.A.: A model for enriching trajectories with semantic geographical information. In: GIS, Seite 22 (2007)

[ABKS99] Ankerst, M., Breunig, M.M., Kriegel, H.P., Sander, J.: Optics: Ordering points to identify the clustering structure, pp. 49–60. ACM Press, New York (1999)

[EKSX96] Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of the 2nd Conference on Knowledge Discovery and Data Mining, pp. 226–231. AAAI Press, Menlo Park (1996)

[Han05] Han, J.: Data Mining: Concepts and Techniques. Morgan Kaufmann Publishers Inc., San Francisco (2005)

[KLK+02] Kalkusch, M., Lidy, T., Knapp, M., Reitmayr, G., Kaufmann, H., Schmalstieg, D.: Structured visual markers for indoor pathfinding (2002)

[Kol33] Kolmogorov, A.N.: Grundbegriffe der Wahrscheinlichkeitsrechnung. Springer, Berlin (1933)

[LFK07] Liao, L., Fox, D., Kautz, H.: Extracting places and activities from gps traces using hierarchical conditional random fields. Int. J. Rob. Res. 26(1), 119–134 (2007)

[NP06] Nanni, M., Pedreschi, D.: Time-focused clustering of trajectories of moving objects. J. Intell. Inf. Syst. 27(3), 267–289 (2006)

[PBKA08] Palma, A.T., Bogorny, V., Kuijpers, B., Alvares, L.O.: A clustering-based approach for discovering interesting places in trajectories. In: SAC 2008: Proceedings of the 2008 ACM symposium on Applied computing, pp. 863–868. ACM, New York (2008)

[SPD+08] Spaccapietra, S., Parent, C., Damiani, M.L., de Macedo, J.A., Porto, F., Vangenot, C.: A conceptual view on trajectories. Data Knowl. Eng. 65(1), 126–146 (2008)

[ZLC+08] Zheng, Y., Li, Q., Chen, Y., Xie, X., Ma, W.-Y.: Understanding mobility based on gps data. In: UbiComp 2008: Proceedings of the 10th Int. Conf. on Ubiquitous computing, pp. 312–321. ACM, New York (2008)