

T-DBSCAN: A Spatiotemporal Density Clustering for GPS Trajectory Segmentation

<http://dx.doi.org/10.3991/ijoe.v10i6.3881>

W. Chen^{1,2}, M. H. Ji¹ and J. M. Wang²

¹ East China Normal University, Shanghai, China

² Tongji University, Shanghai, China

Abstract—Trajectory data generated from personal or vehicle use of GPS devices can be utilized for travel analysis and traffic information service, whereas trip segmentation is a key step toward the semantic labelling of the trajectories. Two issues are difficult to deal with by the traditional density-based algorithms, i. e. multiple stops at the same spatial location with different visit times and non-consecutive point sequence for stop definition due to signal drifting. This article aims to develop a modified density-based clustering algorithm, named T-DBSCAN, by considering the time-sequential characteristics of the GPS points along a trajectory. Two new premises (i.e. state continuity within a single stop and temporal disjuncture among stops) were proposed as a theoretical basis for regulating the trajectory point selection in clustering. An empirical test was performed using a GPS-based personal travel dataset collected in the city of Shanghai to compare T-DBSCAN against DBSCAN. The results indicated that T-DBSCAN effectively improved both accuracy and computational speed in trajectory segmentation.

Index Terms—Personal travel trajectory, Trip segmentation, Density-based clustering, T-DBSCAN

I. INTRODUCTION

The widespread use of GPS-equipped personal travel and vehicle navigation generates huge amounts of trajectory data on a daily basis. As they contain rich information about inner-city travel and spatiotemporal behaviour of urban residents [1], the data have been proven useful for intelligent traffic management and transportation planning [2]. Despite their great application potentials in many fields, the processing of and information extraction from these trajectory data remain a common technical bottleneck. The key issue is how to correctly convert a set of spatiotemporal points (i.e. $S\{s: x, y, z, t\}$) that describe movement to activity information that describes personal travel characteristics (i.e. trip, trip purpose, and travel mode) [3]. This process, usually termed a “semantic annotation of trajectories” [4], needs to be automated in order to support mass data processing.

Trip segmentation is the first and the most important task in trajectory data processing, as its correctness will largely affect such subsequent analyses as OD matrix construction, trip purpose identification, and travel mode detection. As a trajectory is comprised of a series of consecutive moves and stops, a trip can then be defined as a move for a given purpose between two adjacent stops [5]. Therefore, the essential task of trip segmentation is to identify stop points, which are characterized with a num-

ber of travel-related measures that can be derived directly from the trajectory itself. Up to date, several algorithms have been developed and used for trajectory segmentation, including simple time-based clustering, k-means clustering, and spatial density clustering. The purpose of this article is to review the existing major algorithms with respect to trajectory segmentation, identify unsolved issues related to personal travel, and propose a modified density-based method to address these issues. A thorough discussion of the new method and its trajectory segmentation process is provided, which is followed by an empirical test and a comparative evaluation of the new algorithm.

A. Literature Review and Problem Identification

Previous research has mainly identified two ways for stop derivation based on different GPS signal characteristics: (1) comparing temporal gaps between adjacent points along the trajectory to detect stops that are signified by absence of GPS points due to indoor activities [6,7], and (2) applying statistical clustering analysis to identify stops that are signified by a cloud of GPS points confined within a small space due to poor signal reception during outdoor activities [8,9]. The latter case is commonly seen in real applications yet more difficult to deal with, and it is the research focus of this paper.

At present, popular methods for trip segmentation mainly include time-based clustering, K-Means clustering, and density-based spatial clustering [10]. Time-based clustering searches for a point cluster in which the distance between any point pair is less than a distance threshold, d , and their time difference is greater than a given time-interval threshold, t . When found, all points enclosed within the two points are designated as a stop, and the cluster center is calculated. Then, the distance of neighboring points to the cluster center is examined in temporal sequence against the $d/2$ threshold, and those with a distance less than the threshold is labelled as part of the cluster; otherwise, the above procedure is repeated to check for the next stop till the entire trajectory is processed [11,12]. The K-Means clustering for trajectory segmentation is a variant modified from its classic form [13]. It first divides the GPS trajectory with a user-defined number of randomly selected cluster centers (possible stops), then uses a predetermined search radius to re-compute new cluster centers in an iterative manner until the results converge to a satisfactory level (i.e. the cluster centers no longer move from the previous iteration). DBSCAN is a density-based spatial clustering method commonly used for trajectory data processing [14]. It is based on the observation that points surrounding a stop usually form a

high-density cluster. Hereby two parameters are used to define density: the search radius (Eps) and the minimum number of points (MinPts) within a circular area defined by the radius. Points which satisfy some conditions related to density are grouped to form stop clusters (more details in section 3.2). Table 1 shows each of the three methods has its own merits and pitfalls in terms of cluster shape restriction(CS), noise robustness(NR), ease of initial parameter setting(InPara), result certainty(Cert), and computational complexity (CC).

Perusal of the comparative listing in Table 1 indicated that DBSCAN was of overall advantages and especially suitable for irregularly shaped trajectory point clusters. But when processing large-scale spatiotemporal trajectory data, DBSCAN suffers from two drawbacks. First, DBSCAN has a computational complexity of $O(n^2)$, which may cause unbearable time consumption for processing the large amount of daily collected sample data for this study (10,000~30,000 points per day per person). Second, DBSCAN is originally designed for spatial data clustering and handicapped when temporal information is involved. As to the first issue, Ref. [10] proposed a modified algorithm known as DJ-Cluster to simplify DBSCAN's three basic concepts (i.e. direct density reachable, density reachable, and density connected) into a "density-joinable" concept, so that the clusters with common points can be combined in the clustering process. This way, the constraints on computer memories are relaxed to some degree. Another possible solution is to adopt R-tree indexing to reduce the computational complexity [15]; yet none of the currently available open-source software packages with DBSCAN (e.g. R and Weka) supports data indexing. Aiming at the second issue, Ref. [16] brought forward ST-DBSCAN, a DBSCAN variant by indexing spatiotemporal data with R-Tree to explicitly organize observations into temporal neighbours and using two distance thresholds (i.e. Eps1 for spatial and Eps2 for non-spatial) to identify clusters. This approach was mainly designed to process data within an implicit spatial and temporal context; however, it is not suitable for handling GPS-based trajectories, for which the explicit spatial and temporal data are themselves the target of data processing.

When dealing with trajectory data, the situation becomes more complex. One complicated issue is related to repeated visits to an activity location within the same trajectory. As shown in Fig. 1, suppose a person starts his daily travel from home (point B) to office (point S1) in the morning, goes to a restaurant (point S2) for lunch and then back to office (point S3), runs errands at a different location (point S4) in the afternoon and back to office (point S5) again, then travels to another restaurant (point S6) for supper, and finally comes back home (point E) at the end of the day. This example clearly presents a typical human travel pattern that is difficult for traditional density-based clustering methods to deal with, as they will wrongly lump together S1, S3 and S5 as one single stop. In this paper, this type of errors is referred to as a "many-visits-to-one-stop" problem. Apparently, DBSCAN lacks a necessary mechanism to process the temporal sequence of activity stops along the trajectory.

The second limitation of DBSCAN has to do with a stop containing points drifting outside the cluster searching area. As illustrated as S2 in Fig. 1, due to the existence of poor GPS signals, some of the sample points within the

TABLE I.
PROS AND CONS OF THREE MAJOR METHODS FOR TRAJECTORY SEGMENTATION

Method	CS	NR	InPara	Cert	CC
Time-Based	circular	low	Min duration and max range of stop, empirically based	Certain	$O(n)$
K-Means	circular	low	The number of stops (clusters), arbitrarily based	Uncertain	$O(nkt)$
DBSCAN	irregular	high	Point density and density search radius, empirically based	Certain	$O(n^2)$

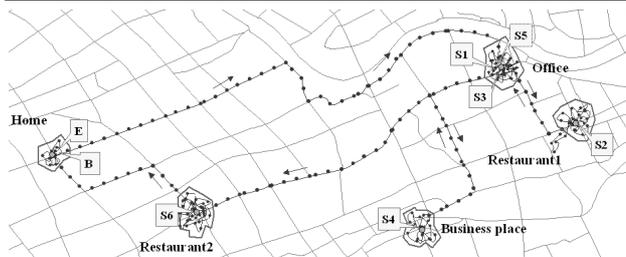


Figure 1. Potential problems associated with the use of DBSCAN for trajectory segmentation

duration of stay at stop 2 drift outside the spatial range of the cluster area. This will lead to misinterpretation of these points as part of a move rather than a stop, causing so-called "spatial disparity of a stop". In this paper, both issues and their combined complexity will be dealt with in the formation of the new algorithm.

II. DEVELOPMENT OF THE T-DBSCAN ALGORITHM

T-DBSCAN (acronym for Trajectory DBSCAN) was developed by extending and modifying the formal definitions of DBSCAN from the basic concepts of GPS based trajectories. Numerous studies unanimously recognize the decomposition of a trajectory into a series of stops and moves; therefore, it is instructive to review these basic concepts in their formal expression first and then provide additional definitions to build the conceptual framework for T-DBSCAN development.

A. Trajectory, Stop, and Move

The conceptual view of a trajectory and its components adopted here is based on the formal work of Ref. [5]. The following three simplified definitions provide a foundation for the work in this paper.

Definition 1: A trajectory is the user defined record of the evolution of the position of an object that is moving in space during a given time interval in order to reach a given destination.

$$\text{trajectory: } [t_{begin}, t_{end}] \rightarrow \text{space}$$

Definition 2: A stop is a part of a trajectory, such that (1) The user has explicitly defined this part to represent a stop, (2) The temporal extent $[t_{beginstop}, t_{endstop}]$ of this part is a non-empty time interval, and (3) The traveling object does not move, i.e. the spatial range of the trajectory for the interval is a single point. (4) All stops are temporally disjoint, i.e. their temporal extents are always disjoint.

Definition 3: a move is a part of a trajectory, such that (1) The part is delimited by two extremities that represent

either two consecutive stops, or t_{begin} and the first stop, or the last stop and t_{end} , or t_{begin} and t_{end} . (2) The temporal extent $[t_{beginmovex}, t_{endmovex}]$ is a non-empty time interval, and (3) The spatial range of the trajectory for interval $[t_{beginmovex}, t_{endmovex}]$ is a spatiotemporal polyline defined by the trajectory function.

B. Definitions for T-DBSCAN

Spaccapietra's definitions clearly indicate some important spatial and temporal characteristics that need to take into account when extracting stops from a trajectory. For the purpose of this paper, these characteristics are expressed as three concepts, i.e. (1) Spatial density – the sample points of one stop are spatially gathering together; (2) State continuity – during the time period of stop, there is no move; and (3) Temporal disjuncture – the temporal extent of two stops are not overlapping with each other. Based on the three concepts, the definitions of the T-DBSCAN algorithm extended from DBSCAN are hereby given below.

Definition 1: Continuous density-based neighborhood (shortened as *CEps&Eps*-neighborhood). Suppose *Eps* is an inner radius defining the density calculation area for a given point taken at time k , p_k , *CEps* is an outer radius for limiting the density searching range, and T is a time interval, during which the distance between p_k and any other point taken at time t , p_t , is less than *CEps*. We then have P , the set of points during T . Then the *CEps&Eps*-neighborhood of p_k , denoted $N_{(p_k)}$, is defined as:

$$N(p_k) = \{q \in P \mid dist(p_k, q) \leq Eps\}$$

where

$$P = T \rightarrow \text{space,}$$

T is a time interval, that satisfies,

$$\text{for } \forall t \in [\min(T), \max(T)], dist(p_t, p_k) < CEps$$

Definition 2: Core point. Let *MinPts* be a threshold number of points within *Eps*. If $N(p_k) \geq MinPts$, then p_k is called a core point. A sample trajectory with a begin point and an end point is shown in Fig. 2. With *MinPts* = 4, point p is designated as a core point since there are four points (in black) within its *Eps* radius comprising its *CEps&Eps*-neighborhood (Fig. 3). The gray points, though also within the same circle, are ruled out since they do not satisfy the “continuous” condition defined by T .

Definition 3: Directly continuous density-reachable (DCDR). If point q is in the *CEps&Eps*-neighborhood of point p , and p is a core point, then q is directly continuous density-reachable from p (Fig. 3). That is, meeting the conditions of $q \in N_{(p)}$ and $N_{(p)} \geq MinPts$.

Definition 4: Continuous density-reachable (CDR). For a spatially ordered set of points p_1, p_2, \dots, p_z , denote $p_1 = n$, $p_z = m$, and n is not DCDR from m . If any p_i is DCDR from p_{i+1} , then n is known as continuous density-reachable from m (Fig. 3).

Definition 5: Continuous density-connectable (CDC). For two points s and t , which are neither DCDR nor CDR from each other, if there is a point o , from which both s and t are CDR, then s and t are known as being continuous density connectable (Fig. 3).

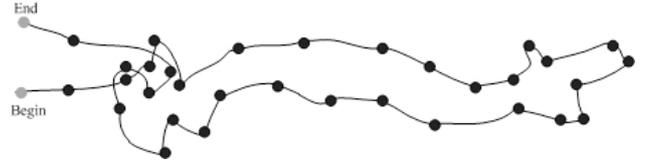


Figure 2. Trajectory example

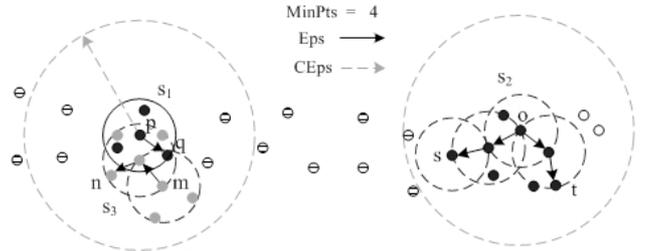


Figure 3. Basic definitions of the T-DBSCAN algorithm

Definition 6: Cluster. Let D be a whole set of the trajectory points. Cluster C is a nonempty subset of D with at least one core point such that, for any two points $\forall p, q \in D$, (1) $q \in C$ if $p \in C$ and q is CDR from p or (2) $\forall p, q \in C$ if p and q are CDC from each other. Examples are illustrated as the gray and dark points in Fig. 3.

Definition 7: Temporally continuous (TC). Let the min and max time stamps of a cluster $C (\subseteq D)$ be $mint$ and $maxt$, respectively. C is known as being “temporally continuous” if, for $\forall p_t \in D$ and $mint < t < maxt$, $p_t \in C$.

Definition 8: Temporally overlapping (TO). Let two clusters $C_1 (\subseteq D)$ with a time extent $t_{[mint1, maxt1]}$ and $C_2 (\subseteq D)$ with $t_{[mint2, maxt2]}$. C_1 and C_2 are known as being temporally overlapping, if $t_{[mint1, maxt1]} \cap t_{[mint2, maxt2]} \neq \emptyset$.

Definition 9: Stop. Let C_1, C_2, \dots, C_n be n clusters in D . Stop S results as a nonempty subset of D from the following situations: (1) C_i is a stop if it is TC and not TO with any other cluster; (2) For $\forall C_i$, if C_i is not TC but the non-TC points (blank points in Fig. 3) are within its time extent, C_i is still treated as a stop, and the non-TC points are merged to C_i as part of the stop; (3) For $\forall C_i, C_k$, if C_i and C_k are TA to each other, then a stop is formed by merging C_i and C_k .

III. DEVELOPMENT OF T-DBSCAN

The new definitions, i.e. “temporal continuous” and “temporal overlapping”, are intended to express the spatial and temporal rules for identifying true stops from a trajectory. T-DBSCAN is so designed to overcome the problems of “many-visits-to-one-stop” and “spatial disparity of a stop” that cannot be handled by the traditional DBSCAN method. Its development process hereby consists of an overall workflow, a search method for continuous density-based neighborhood, and a cluster expanding method.

A. T-DBSCAN Workflow

Except the last step, T-DBSCAN follows roughly the same general workflow as DBSCAN. It first creates clus-

ters by searching the continuous density-based neighborhood of qualified core points, then expands the clusters based on the core point and its neighboring points, and finally merges the clusters which are temporally adjacent to each other. More differences are presented in the details of the algorithm. As shown in the following pseudo code, four input parameters are required for T-DBSCAN: D is the set of points comprising the trajectory; $CEps$ is the distance range to ensure that the points comprising a stop are of state continuity (that is, to rule out the points that belong to a “move” during the time extent of the stop); Eps is the search radius for identifying density-based neighborhood; and $MinPts$ is the minimum number of neighboring points to identify a core point.

```

T-DBSCAN (D, CEps, Eps, MinPts)
  C = 0 // id number of the cluster currently being searched
  MaxId = -1 //the maximum id of the visited point
  /*Because the points of a trajectory are time ordered, the points
  with an id smaller than MaxId are supposed to either belong to
  the previous cluster (the points unvisited but during the time
  extent of a cluster also belong to the cluster based on definition
  7) or identified as noise. So here only the points with an id
  greater than MaxId are processed, and this helps to improve the
  efficiency of the method.*!
  for each P > MaxId in dataset D
    mark P as visited
  //search for continuous density-based neighbors, N is the set of
  neighbors
  N = getNeighbors (P, CEps, Eps)
  MaxId = id of P
  if sizeof(N) > MinPts
    C = C+1
  endif
  //expand the cluster based on definition 3, 4, and 5; Cp is the
  point set of the cluster with id of C
  (Cp,MaxId) = expandCluster(P, N, Eps, MinPts, MaxId)
  endfor
  //process the clusters into stops based on (3) of definition 9
  for each cluster
    if max point id of cluster_i >= min point id of cluster_i+1
      else merge cluster_i and cluster_i+1
    endif
  endfor

```

B. Searching for Continuous Density-based Neighbourhood

Fig. 4 illustrates how a continuous density-based neighborhood is found with T-DBSCAN. Since GPS points of a trajectory are time ordered, in most cases two consecutive points in time sequence are also spatially most adjacent. In Fig. 4, let p be the target point. A forward search in time sequence is first performed to identify and collect continuous density-based neighboring points. If the distance between p and a candidate point is less than Eps , then the point is labelled as a neighbor of p , else if the distance is greater than $CEps$ then the searching process stops. This results in the grey points as the neighbors of p . And the hollow ones are separated from p by the points that indicating a move.

Here, $CEps$ is a key parameter to avoid the above two groups of points to be lumped into one cluster. Its physical significance is: when the GPS carrier stops or moves slowly, the signal quality of GPS might become poor, causing the GPS track points to drift. Here, parameter $CEps$ takes the experimental value of maximum drift range. If the distance of two points is beyond the range, it is reasonable to believe that there is a move, so the cur-

rent point and the core point cannot be in the same cluster. The pseudo code of the continuous density-based neighborhood searching process is as follows.

```

N = getNeighbors(P, CEps, Eps)
// searching forward for neighbors
for each P' when id of P' > id of P
  if distance(P',P) < Eps
    Add P'into N
  else distance(P',P) > CEps
    break
  endif
endfor

```

C. Cluster Expanding Process

As previously defined, a new cluster is created when point p is qualified as a core point, which is in turn defined by its neighbouring points being equal to or greater than $MinPts$. The cluster expanding process proceeds as follows. Each neighbour of the core point is traversed to examine its quality for being a core point until all CDR and CDC points are examined. The final cluster is then formed with interconnected core points and their neighbouring points, as well as their CDR and CDC points.

```

(Cp,MaxId) = expandCluster (P, N, Eps, MinPts, MaxId)
add P to cluster Cp
for each point P' in N
  mark P' as visited
  if id of P' > MaxId
    Maxid = id of P'
  endif
  // find the neighbors of neighbors of core point P
  N' = getNeighbors(P', CEps, Eps)
  if sizeof(N') >= MinPts
    // classify the points into current cluster based on defini-
    tion 3,4,5
    N = N joined with N'
  endif
  if P' is not yet member of any cluster
    add P' to cluster Cp
  endif
endif
endfor

```

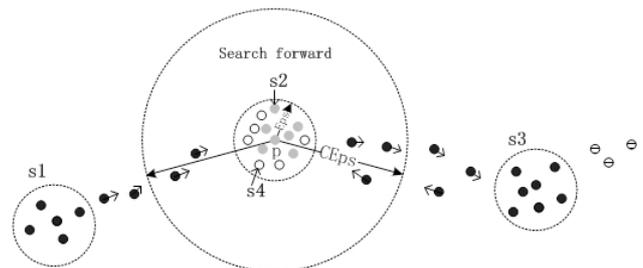


Figure 4. Searching method of continuous density-based neighborhood

IV. TESTS AND COMPARISONS

Two tests were performed to provide a comparative analysis between DBSCAN and T-DBSCAN in terms of segmentation accuracy and computation efficiency. One-day GPS travel trajectories collected from 50 volunteers in Shanghai were used as the test dataset, which contained such essential data items as point ID, timestamp, and location in 3-dimensional space. A laptop computer with a 2.4 GHz CPU, 2 GB memory, and Windows XP was used to perform the analysis. In the tests, both algo-

rithms used the same parameter setting, i.e. $MinPts = 120$, $Eps = 20$ m, and $CEps = 300$ m.

The first test was designed to compare computational efficiency of the two algorithms at different levels of data volume. The trajectories were first sorted and grouped into three classes of data volume (Table 2) before segmentation processing. The processing time was recorded for either algorithm per trajectory, and the average amount of processing time was summarized for each volume class for either algorithm separately (Table 2). It indicated that T-DBSCAN was significantly faster than DBSCAN in segmenting the trajectories at all data levels, and the efficiency improvement seemed to ascend with the data volume to be processed. Specifically, T-DBSCAN took only 6.25% of DBSCAN's time to process up to 8000 points, and this ratio decreased to 5.72% and further 4.34% when 8000~20000 points and 20000~30000 points were processed, respectively.

The second test involved comparison of segmentation accuracy between the two methods. One trajectory was chosen for demonstration here, and its geometry and actual activity stops were illustrated in Fig.5. Excluding the start and end points (symbolized as red bubbles in the figure), the entire trajectory contains a total of nine stops (represented by blue bubbles). The number in each blue bubble indicates the order of the stop being visited along the one-day trajectory. As a matter of fact, some locations were visited more than once, hence forming the so-called "many-visits-to-one-stop" problem. The spatially-tangled but temporally separate clusters were properly decomposed by T-DBSCAN, as there was no overlapping between point ID ranges (i.e. not TO) of any pair of adjacent clusters (Table 3). Compared to the real stops from field verification, all T-DBSCAN-deduced clusters had correctly a match except cluster 5, though spatiotemporally so well qualified but only found to be resulting from a traffic jam. In comparison, serious overlapping occurred between many pairs of clusters identified with DBSCAN, such as clusters 1 and 2, 6 and 7, as well as 9 and 10. Because of the serious overlapping resulting from the convoluted trajectory, these clusters could hardly match with any of the real stops in sequence.

The spatial disparity problem, a second issue associated with segmenting a travel trajectory, was also adequately dealt with in the T-DBSCAN processing, as the algorithm was able to automatically snap the temporally-overlapping (note: including "temporally-touching") clusters together to form one single stop (e.g. stops 3, 5, 7, and 8). As a result, the total number of stops identified using T-DBSCAN basically matched the number of real stops from field verification, whereas the stops produced from DBSCAN could not be easily fit to the real observations.

V. CONCLUSIONS

Personal travel trajectories as recorded by a GPS device can simply be as complex as the daily spatiotemporal behavior of any individual human being. Convoluted trajectories may contain the so-called "many-visits-to-one-stop" and "spatial disparity of a stop" problems. Trip segmentation of a GPS-induced trajectory involves properly utilizing inherent information of not only the spatial distribution of sample points and related geometric features, but also their temporal sequence. As demon-

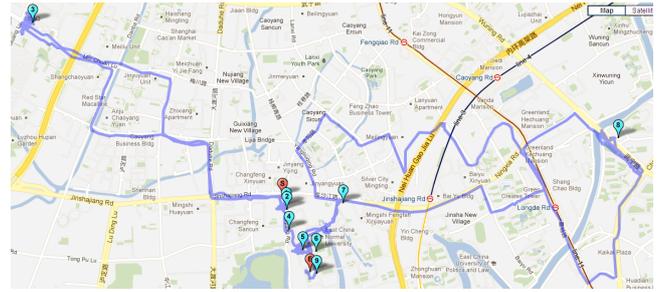


Figure 5. The geometry of and activity stops along the selected trajectory

TABLE II. COMPARISON OF COMPUTATIONAL EFFICIENCY BETWEEN DBSCAN AND T-DBSCAN

Class	Number of Points	Processing Time	
		DBSCAN	T-DBSCAN
1	0~8000	16s	1s
2	8000~20000	70s	4s
3	20000~30000	369s	16s

TABLE III. COMPARISON OF TRAJECTORY SEGMENTATION ACCURACY BETWEEN DBSCAN AND T-DBSCAN

Method	Sequence of clusters being identified (represented as ClusterID: {PointID1, PointID2})			
DBSCAN	1:{3, 7602}; 2:{620, 7533}; 3:{1721, 6368}; 4:{6509, 6646}; 5:{7770, 8155}; 6:{8270, 16441}; 7:{8346, 23200}; 8:{16907, 22652}; 9:{18580, 21115}; 10:{18740, 18876}; 11:{16761, 22974}; 12:{23401, 25140}			
	Identified Cluster	Real Stop	Identified Cluster	Real Stop
	1:{3, 593}	Stop 1	7:{8270, 15911} 8:{15912, 16336}	Stop 5
	2:{622, 821}	Stop 2	9:{16442, 16592}	Stop 6
T-DBSCAN	3:{1721, 5694} 4:{5695, 6154}	Stop 3	10:{16907, 17039} 11:{17040, 17876}	Stop 7
	5:{6509, 6630}	No stop	12:{18580, 19805} 13:{19806, 20098} 14:{20099, 21111}	Stop 8
	6:{7770, 8129}	Stop 4	15:{23401, 25025}	Stop 9

demonstrated in this study, traditional density-based clustering methods were proven both theoretically and practically incapable of identifying multi-visited stops, as they only consider the spatial characteristics of a trajectory. The design of T-DBSCAN was based on comprehensive consideration of spatiotemporal characteristics of the trajectory. By doing so, the new algorithm was able to untangle convoluted travel patterns through proper treatment of the state continuity and temporal disjuncture issues, leading to a higher accuracy in trip segmentation. In addition, T-DBSCAN was proven computationally more efficient and therefore suitable for mass data processing. On the other hand, running T-DBSCAN requires more input parameters, and they are empirically based and therefore necessarily difficult to determine. This issue must be dealt with in the future work.

REFERENCES

- [1] K. W. Axhausen, S. Schönfelder, J. Wolf, M. Oliveira, and U. Samaga, "Eighty weeks of GPS traces: approaches to enriching the trip information," In *Proc. 83th Annual Meeting of the Transportation Research Board*. Washington, D.C., January, 2004.
- [2] A. Pinjari, N. Eluru, R. Copperman, I. N. Sener, and J. Y. Guo, "Activity-based travel-demand analysis for metropolitan areas in Texas: CEMDAP models, framework, software architecture and application results," Research Report 4080-8. Center for Transportation Research, University of Texas at Austin, 2006.
- [3] P. R. Stopher and S.P. Greaves, "Household travel surveys: where are we going?" *Transportation Research Part A: Policy and Practice*, vol. 41, no.5, pp.367-381, 2007. <http://dx.doi.org/10.1016/j.tra.2006.09.005>
- [4] B. Guc, M. May, Y. Saygin, and C. Körner, "Semantic annotation of GPS trajectories", In *Proc. 11th AGILE International Conf. Geographic Information Science 2008*, Girona, Spain, 2008.
- [5] S. Spaccapietra, D. Chakraborty, K. Aberer, C. Parent, and Z. Yan, "A conceptual view on trajectories," *Data Knowl. Eng.*, vol. 65, no. 1, pp.126-146, 2008. <http://dx.doi.org/10.1016/j.datak.2007.10.008>
- [6] P. R. Stopher, "GPS, location and household travel," *Handbook Transport Geogr. Spatial Syst.*, vol. 5, pp. 433-449, 2004.
- [7] J. Du and L. Aultman-Hall, "Increasing the accuracy of trip rate information from passive multi-day GPS travel datasets: automatic trip end identification issues," *Transport. Res. A: Pol.*, vol. 41, no.3, pp. 220-232, 2007.
- [8] S. T. Doherty, N. Noel, M. Lee-Gosselin, C. Sirois, M. Ueno, F. Theberge, "Moving beyond observed outcomes: integrating global positioning systems and interactive computer-based travel behaviour surveys," *Transport. Res. Circ.*, vol. E-C026, pp. 449-466, 2001.
- [9] P. R. Stopher, Q. Jiang, and C. FitzGerald, "Processing GPS data from travel surveys," In *Proc. 2nd International Colloquium on the Behavioural Foundations of Integrated Land-use and Transportation Models: Frameworks, Models and Applications*. Toronto, June, 2005.
- [10] C. Zhou, D. Frankowski, P. Ludford, S. Shekhar, and L. Terveen, "Discovering personally meaningful places: an interactive clustering approach," *ACM T. Inform. Syst. (TOIS)*, vol. 25, no. 3, pp. 459-466, 2007. <http://dx.doi.org/10.1145/1247715.1247718>
- [11] H. Ramaswamy and T. Kentaro, "Project lachesis: parsing and modeling location histories," In *Proc. GIScience 2004*, vol. 3234, pp.106-124, 2004.
- [12] J. Liu, O. Wolfson and H. Yin, "Extracting semantic location from outdoor positioning systems," In *Proc. 7th Int. Conf. Mobile Data Management*, Nara, Japan, 2006.
- [13] D. Ashbrook and T. Starner, "Learning Significant Locations and Predicting User Movement with GPS," In *Proc. IEEE Sixth Int. Symposium on Wearable Computing*, Seattle, WA, USA, pp. 101-108, 2002. <http://dx.doi.org/10.1109/ISWC.2002.1167224>
- [14] M. Ester, H. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," In *Proc. 2nd Int. Conf. Knowledge Discovery and Data Mining*, Portland, Oregon, USA, pp. 226-231, 1996.
- [15] A. Guttman, "R-trees: a dynamic index structure for spatial searching," In *Proc. 1984 ACM SIGMOD Int. Conf. Management of Data*, ACM New York, NY, USA, vol. 14, no. 2, pp. 47-57, 1984.
- [16] D. Birant and A. Kut, "ST-DBSCAN: an algorithm for clustering spatial-temporal data," *Data Knowl. Eng.*, vol. 60, no. 1, pp. 208-221, 2007. <http://dx.doi.org/10.1016/j.datak.2006.01.013>

AUTHORS

W. Chen is with Shanghai Research Center for Spatial Information and GNSS, East China Normal University, Shanghai 200241, China (e-mail: chenwen_ecn@126.com).

M. H. Ji is with Key Lab of GIScience, Education Ministry of China, East China Normal University, Shanghai, 200241, China (e-mail: mhji@geo.ecnu.edu.cn).

J. M. Wang is with College of Surveying and Geoinformatics, Tongji University, Shanghai, 20092, China (e-mail: jianmeiw@tongji.edu.cn).

This work is sponsored by the Open Research Fund of the Key Laboratory of Embedded System and Service Computing No 48505280, Natural Science Foundation of China (No. 41201380, No. 40771138 and No.61372086) and the Specialized Research Fund for the Doctoral Program of Higher Education (20120072120057). Submitted 15 May 2014. Published as resubmitted by the authors 25 October 2014..