Investigating the Travel Pattern of Threatened Individuals – An Innovation for CBI and Interpol

Samuel P, Prakash S, and Srihari G,
Department of Computer Science and Engineering,
Kings Engineering College,
Chennai 602117.

Abstract—Frequent route is an important individual outdoor behavior pattern that many trajectory-based applications rely on. In this paper, we propose a novel framework for extracting frequent routes from personal GPS trajectories. The key idea of our design is to accurately detect road corners and utilize these new metaphors to tackle the problem of frequent route extraction Concretely, our framework contains three phases: 1) characteristic point (CP) extraction; 2) corner detection; and 3) trajectory mapping. In the first phase, we present a linear fitting-based algorithm to extract CPs. In the second phase, we develop a multiple density level DBSCAN (density-based spatial clustering of applications with noise) algorithm to locate road corners by clustering CPs. In the third phase, we convert each trajectory into an ordered sequence of road corners and obtain all routes that have been traversed by an individual for at least *F* (frequency threshold) times. We evaluate the framework using real-world trajectory datasets of individuals for one year and the experimental results demonstrate that our framework outperforms the baseline approach by 7.8% on average in terms of precision and 21.9% in terms of recall.

Index Terms—Characteristic point extraction (CPE), corner detection, frequent routes.

I. INTRODUCTION

WITH the wide adoption of GPS receivers in vehicles and smartphones, huge amounts of personal GPS trajectories have been accumulated. By analyzing those GPS trajectories, we can understand each individual's mobility patterns and obtain valuable insights about her/his daily behavior. These patterns and behaviors can be further utilized to improve the quality of various trajectory-based services, such as route prediction, disorientation detection, trip planning, and location-based recommendation.

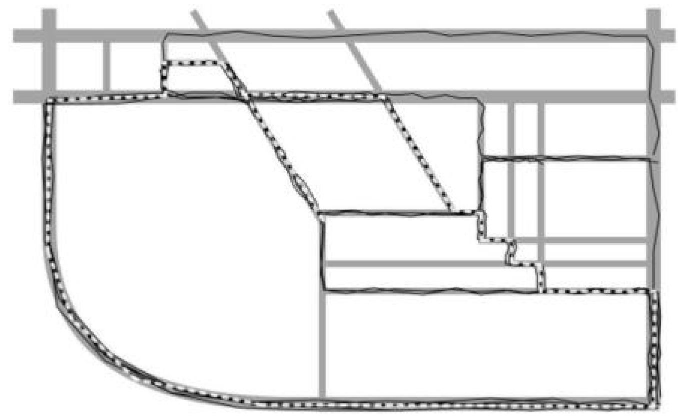

Fig. 1. Illustration of frequent routes.

Frequent route is an important individual outdoor behavior pattern that the aforementioned ubiquitous applications rely on. Fig. 1 illustrates an example of frequent routes. In the figure, gray lines denote the physical roads and black lines denote one individual's GPS trajectories. The white dotted lines highlight the frequent routes of the individual's outdoor movements. In the example, we define a route to be a frequent route only if an individual has traversed the route for a certain amount of time. There are some existing works that attempt to extract frequent routes from personal GPS trajectories. However, extracting frequent routes from personal GPS trajectories is still a challenging problem for the following reasons. 1) GPS readings are not accurate due to hardware constraints. Inaccurate GPS readings and frequent speed changes in a trip result in irregular fluctuations in GPS trajectories. Mathematically, it is difficult to define an accurate distance function to measure trajectory similarity for irregular fluctuated trajectories. Without timing information in trajectories, this problem is even more difficult. Thus, the frequent route extraction methods based on trajectory similarity cannot be applied to irregular fluctuated trajectories. 2) Physical roads are different from each other. Some direction changes are sharp, while others are smooth. Thus, it is difficult to accurately define where a road direction changes. This ambiguous feature of roads naturally propagates to GPS trajectories and disables those frequent route extraction methods based on segmenting and clustering trajectories since they need to partition trajectories at special GPS points where trajectory direction changes. 3) Different physical roads have different trajectory densities due to variation in visit instances collected. This fact indicates that existing frequent route extraction methods based on clustering trajectories using uniform trajectory density cannot reliably detect all the trajectory clusters with different densities. 4) The ideal route representation should be concise and close to the corresponding physical roads. However, existing frequent route extraction methods based on clustering trajectories either use multiple points to represent a physical road segment between two road corners, or use simple direct connections between two hot regions to represent physical road segments. To tackle the above challenges, we conduct preliminary analysis on a large number of personal GPS trajectories and obtain the following three observations .1) Individuals' daily outdoor movements are constrained by physical roads. 2)

Physical roads' topology information, such as road corners, is embedded in personal GPS trajectories. 3) If road corners can be detected accurately, the physical roads can be most concisely represented by connecting all the road corners sequentially. Based on these observations, we propose a novel frequent route extraction framework that leverages corner detection techniques for making full use of physical road topology information embedded in personal GPS trajectories. Particularly, instead of defining complicated similarity metrics for clustering GPS trajectories, our method maps GPS trajectories onto physical roads with the aid of corner detection and outputs concise frequent routes in the form of physical road segments. We validate our framework on a large number of personal GPS trajectories. Our main contributions are summarized as follows. 1) To the best of our knowledge, we are the first to leverage physical road topology information, particularly road corners and connectivity between them, for frequent route extraction. 2) We propose a characteristic point extraction (CPE) method based on linear fitting techniques. The CPE method filters out irregular fluctuations in GPS trajectories and identifies actual characteristic points (CPs) at road corners with different sharpness. 3) We design a multiple density level density-based spatial clustering of applications with noise (DBSCAN) (MDL-DBSCAN) algorithm based on an existing algorithm to detect road corners by clustering CPs with different densities. 4) We define each trajectory as an ordered sequence of detected road corners rather than grid cells. This trajectory representation streamlines our cluster fusing process, which maps trajectory clusters onto physical roads. 5) We evaluate the proposed framework with real-world GPS trajectories collected from more than 6800 individuals for a year. Experimental results demonstrate that our framework outperforms the trajectory clustering-based method as a baseline method in terms of both precision and recall. The rest of this paper is organized as follows.

II. OUR PROPSED FRAMEWORK

In this section, we present the design of our frequent route extraction framework.

A. *Frequent Route Extraction Framework*

Based on the observations 1) and 3) in Section I, we convert the frequent route extraction problem into a traversed road segment counting issue. By mapping personal GPS trajectories onto physical roads, we need to determine the road segments contained in each trajectory and count the number of times each road segment has been traversed. If the number of traversed time for a route is greater than a certain threshold F, then the route is identified as the frequent route for an individual. To solve the aforementioned problem, we have to solve two sub-problems: 1) how to extract roads topology information to segment a GPS trajectory and 2) how to map GPS trajectories onto road segments in physical roads. According to the observations 1) and 2) in Section I, human outdoor movements are constrained by physical roads. Thus, the GPS trajectories of personal movements inevitably contain the connected road segment of physical roads. Road corners are identified as the metaphors to separate road segments in GPS trajectory. Physically, it is reasonable to infer that the locations, at which a number of trajectories direction change, are road corners. To utilize road corners for frequent route extraction, we can first identify the special points in each trajectory, where trajectory direction changes significantly and steadily. Then, we can locate road corners by clustering these special points. If a trajectory traverses two corners successively, the two corners form a road segment that has been traversed. By locating road corners in trajectories, we generate a minimal number of points to separate a GPS trajectory into a sequence of road segments, we then extract frequent routes as the road segments that have been traversed more than F times by an individual.

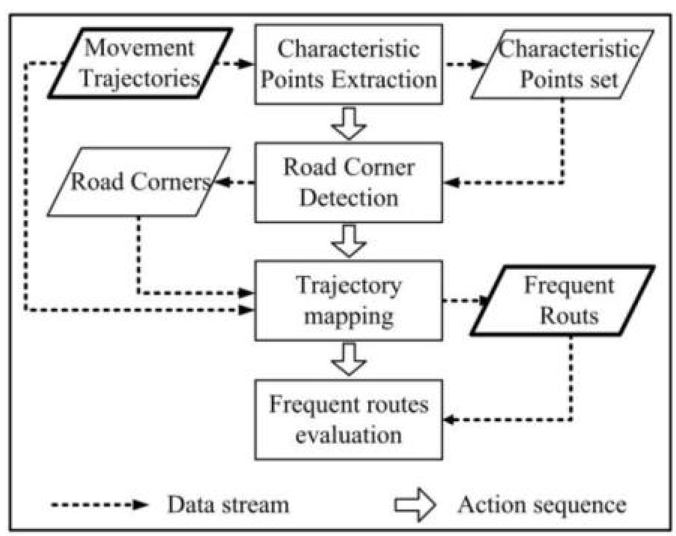

Fig. 2. Overview of our framework.

With the road corners, the second sub-problem of mapping trajectories into road segments in physical roads is equivalent to transforming each trajectory into an ordered sequence of road segments defined by road corners. Based on the above analysis, we propose a novel framework to extract frequent routes. Fig. 2 illustrates the overview of our framework. Generally, the framework consists of four steps. In the first step, a CPE method is applied to extract the CPs in each trajectory. In the second step, an improved DBSCAN clustering algorithm named MDL-DBSCAN is used to locate the road corners. In the third step, all the trajectories are mapped onto physical roads and the frequent routes are extracted. Finally, the effectiveness of frequent route extraction framework is evaluated using a real-world trajectory dataset.

B. Characteristic Point Extraction

As shown in Fig. 2, given a large collection of GPS trajectories, the first task is to extract CPs of each trajectory. As mentioned previously, a CP is the GPS point where the trajectory's direction changes significantly and steadily (we will present its formal definition later). Fig. 3 illustrates an example of individual movement trajectory denoted as a gray line on the right panel. This trajectory is constrained by physical roads apparently. The CPs, marked as points, are the locations where trajectory direction changes significantly and steadily. Given a GPS trajectory $Ti = p1p2 \dots pni$, an intuitive method to extract the CPs is measuring the angle between segment $pj{-}1pj$ and $pjpj{+}1$. Unfortunately, in practice, GPS trajectory often suffers the low-sampling-rate problem, i.e., GPS devices collect data at a low and unstable frequency. What's worse is that GPS drift error cannot be ignored. Due to low sampling rate and drift errors, irregular fluctuations exist in GPS trajectories. An example is illustrated in the left panel in Fig. 3. If we adopt the intuitive idea directly, the irregular fluctuation in GPS trajectory will result in poor results. To circumvent the problem, we introduce a method based on linear fitting. Fig. 4 depicts an example. To detect the direction change at GPS point pj, we first linear fit the point sets

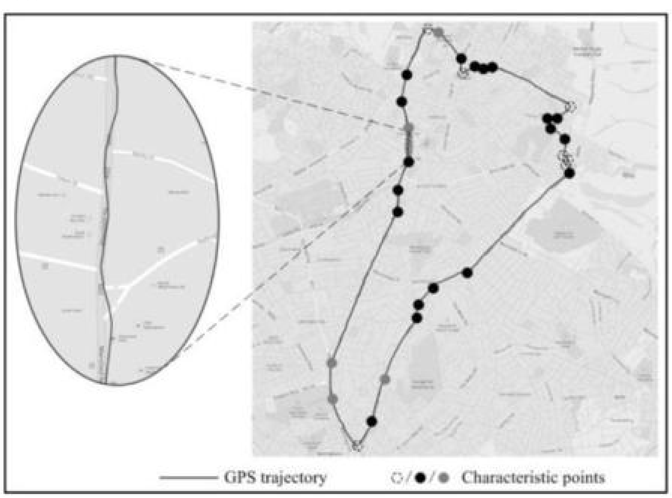

Fig. 3. Illustrative example of CP.

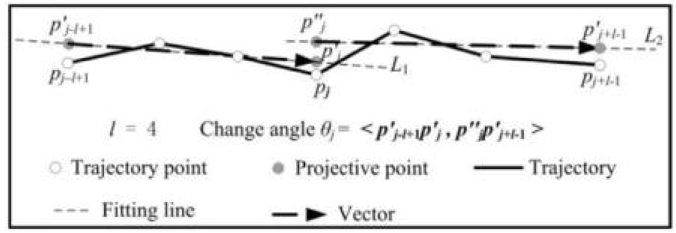

Fig. 4. Illustrative example of direction change angle detecting based on linear fitting.

$\{ pj{-}l{+}1, pj{-}l{+}2, \dots, pj\}$ and $\{ pj, pj{+}1, \dots, pj{+}l{-}1\}$ and obtain two straight lines $L1$ and $L2$. The referenced parameter l denotes the fitting length, i.e., the number of points the linear fitting method considers on each side of pj.

Algorithm 1 CPE Method

Input: trajectory $T = p1p2 \dots pj \dots pn$; change angle threshold $\theta thrd$;
; change angle decrease rate r; linear fitting length l
Output: *CP* set *CPs ;*
lastCPP ← *l*-1;
for *j* from *l*-1 to *n-l* **do**
if *isCPP(pj, l, θthrd)* **then** // check if *pj* is a *CPP*
Add *pj* into *CCPs*; // *CCPs* denotes the set of *CCP*
lastCPP ← *j*;
else
if *isSlowChange*(*lastCPP*, *j*, *l*, *θthrd*) **then**
while *flag* **do**
θthrd = *θthrd**(1-*r*);
for *index* from *lastCPP* + 1 to *j* **do**
if *isCPP*(*pj, l, θthrd*) **then**
Add *pj* into *CCPs*;
lastCPP ← *j*;
flag ← *false*;
end for
end while
end for
Extract *LCCSs* from *CCPs*;
for each *S* in *LCCSs* **do**
get the point *p* whose change angle is largest among *S*;
add *p* into *CPs*;
end for
add *p*1 and *p*n to *CPs*. //add the start and end point to *CPs*

TABLE I
EVALUATION OF THREE CP EXTRACTION RESULTS

Method	Precision	Recall
MDL based method	1	0.625
Linear Fitting based method	1	0.656
CPE method	0.939	0.969

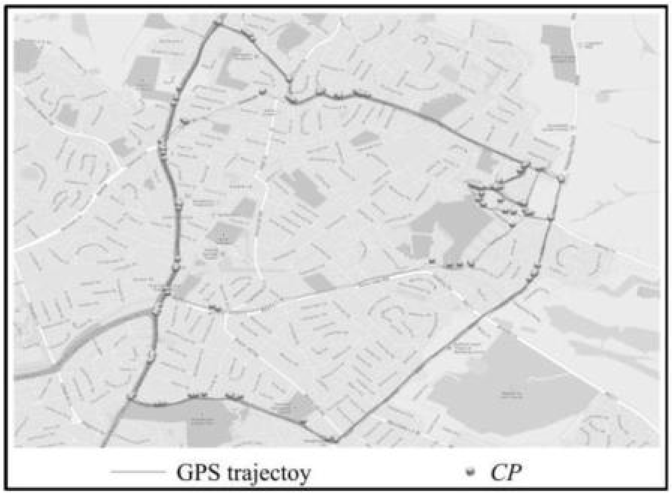

Fig. 5. CPs extraction result by CPE on a trajectory set that contains nine trajectories.

We can see that although the precision of CPE method is slightly lower than the other two methods, its recall is much higher than the other two methods. The CPE method achieves almost 97% detection rate with the false alarm rate 6.1%. Compared to the manually labeled CPs in Fig. 3, we found that the MDL principle-based method is able to extract the CPs in the slow direction change regions, but it ignores the CPs in the regions where there are relatively small but continuous "*S*"- or "*Z*"-shaped direction changes. The method based on linear fitting cannot find the CPs in slow direction change regions [shown as the light-gray rectangle shadow in. In contrast, the CPE method is superior to the above two methods, since it can not only find the CPs in slow and sharp direction change regions simultaneously, but also catch the CPs in regions that are relatively small but have continuous *S*- or *Z*-shaped direction changes.

C. Road Corner Detection

In our framework, after extracting CPs, our next task is to detect road corners. Based on the observation 1) in Section I, personal outdoor movement trajectories are restricted by physical roads. Conversely, personal movement trajectories contain physical clues that reflect physical roads. We observe that the CPs form several clusters and these clusters match the road corners perfectly. Based on this observation, we assume that the locations, which contain a number of trajectories direction changes, are road corners. Thus, in our framework we detect road corners by clustering CPs, since CPs capture direction changes. To cluster CPs, we have the following four requirements for the clustering algorithm.

1) The algorithm should be able to identify the number of clusters automatically. It is impractical to know the number of road corners of a given trajectory set in advance. Thus, the algorithm should figure it out.

2) The algorithm should be able to find CP clusters with different shapes, since CP clusters around corners usually have different spatial shapes.

3) The algorithm should be able to eliminate the "noise" automatically.

In this paper, we ignore isolated CPs and clusters containing a few CPs since they will not lead to frequent route.

4) The algorithm should be able to find CP clusters with different densities. Due to different quantities and distribution shapes, the CP clusters always have different densities.

Existing density-based clustering algorithms, such as DBSCAN and OPTICS, can adapt to the first three requirements. However, to the best of our knowledge, none of the clustering algorithms can adapt to the fourth requirement. In this paper, we propose MDL-DBSCAN algorithm that can meet not only the first three requirements but also the fourth requirement. DBSCAN and OPTICS employ a unique global density threshold in the clustering process. It is the primary reason why they cannot detect the clusters with different densities. Unlike DBSCAN or OPTICS, MDL-DBSCAN clusters the CPs with multiple density thresholds. In MDL-BSCAN, DBSCAN works as a sub-procedure to cluster CPs with a given density threshold. The core techniques of MDL-DBSCAN can be summarized as "one process" and "two constraints." *1) One Process:* To extract the CP clusters with different densities, MDL-DBSCAN clusters CPs on MDL iteratively. *2) Two Constraints:* In order to control DBSCAN to perform as the process shown in Fig. 8, we have to add two constraints additionally. a) The one process may merge the clusters generated at previous density levels together when the current density level is low enough. Thus, we add the following constraint. *Constraint 1:* The clusters generated at previous density levels (i.e., high-density levels) cannot be partitioned or merged into other clusters as density level decreased (i.e., lower density levels). Additionally, we do not forbid the extension of the clusters generated in previous density levels, since it may result in many meaningless small clusters. b) At each iteration of DL-DBSCAN, the unlabeled CPs that satisfy current density level may be merged into the clusters generated at previous density levels although they can independently form a new clusters at the current density level. Thus, we add the following constraint. *Constraint 2:* The unlabeled point clusters that satisfy the current density level (i.e., the density level, on which, one process is clustering the CPs) should independently form a new cluster instead of merging into the previously generated clusters.

Algorithm 2 CPs Clustering Algorithm

Input: *CP* list *CPl*; *HighestDensity*; *LowestDensity* ; *DensityLevel*

Output: *CPs* with cluster label*;*

1: *startID* ← 0;

2: **for** *j* from 1 to *DensityLevel* **do**

3: Compute the density threshold *Densityj = (Epsj,MinPtsj)*;
/*handling the unlabeled points preferentially */

4: Move the unlabeled *CPs* to the beginning of the *CPl*;
/*using the *DBSCAN*, the *neighbors* definition of which has been */
/*modified according to Definition 4, to cluster the *CPs* in *CPl* */

5: *startID* ← *DBSCAN(CPl, Epsj, MinPtsj, startID* + 1*)*;

/* let all the *CPs* can be visited again */
6: Change the *VisitedFlag* of each *CP* to *false*;
7: **end for**

D. Trajectory Mapping

In our framework, the next task is to map each trajectory onto physical roads and construct connectivity matrix (CM) among all involved corners. Then, we can get frequent routes by retrieving CM with a given frequency threshold.

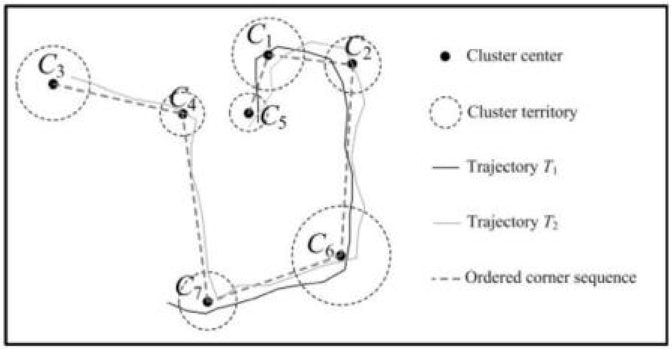

Fig. 6. Example of trajectory mapping.

For each trajectory, the trajectory mapping procedure checks it from its starting point to the end point. If it traverses one cluster territory, the center of this cluster will be added into the ordered sequence of this trajectory. Since each cluster center corresponds to a road corner, the trajectory mapping procedure actually transforms each original trajectory into an ordered sequence of road corners. From the ordered sequences, we can obtain the connectivity information among involved road corners and the number of trajectories mapped onto each physical road segment.

E. Frequent Route Evaluation Methods

We evaluate the extracted frequent routes from two aspects. First, we measure the closeness from them to the corresponding physical roads. Second, by using the manually labeled frequent routes as ground truth, we analyze their precision and recall. *1) Closeness to Physical Roads:* How close the extracted frequent routes are to the physical roads is an important indicator to evaluate our framework. The ideal method of measuring the distance between physical roads and extracted frequent routes should compute the average point-to-line distance from the sample points of physical roads to the corresponding extracted frequent routes. However, the ideal method is infeasible since the numerical information of the involved roads is not available from electronic map applications such as Google Maps, and we can only obtain maps in the form of pictures. Therefore, we employ an approximate method, which measures the average point-to-line distance from the sample points of GPS trajectories to the corresponding frequent routes.

III. CONCLUSION

Frequent routes are an important context for trajectory-based applications. In this paper, we propose a novel framework to extract frequent routes from personal GPS trajectories. In our framework, we first propose a CPE method to extract CPs, which characterize the physical roads. Second, we propose the MDL-DBSCAN clustering algorithm to locate road corners. Instead of using a unique global density threshold, MDL-DBSCAN uses several density thresholds to cluster the CPs at MDL. Third, we propose a method to map all the trajectories on physical roads and detect frequent routes. Finally, we evaluate our framework on real personal GPS trajectory datasets. The results demonstrate that our framework outperforms the baseline approach by 7.8% in terms of average precision and 21.9% in terms of average recall. In the future, we plan to extend this paper in two directions. First, we attempt to exploit other information embedded in GPS trajectories such as time stamps for frequent route extraction. Second, we intend to develop practical applications, such as disorientation detection and movement trend prediction, leveraging the techniques developed in this framework.

REFERENCES

[1] Hongyan Liu Jiawei Han Dong Xin Zheng Shao, "Mining Frequent Patterns from Very High Dimensional Data:A Top-Down Row Enumeration Approach", Department of Management Science and Engineering, Tsinghua University hyliu@tsinghua.edu.cn, Department of Computer Science, University of Illinois at Urbana- Champaign {hanj, dongxin, zshao1}@uiuc.edu.

[2] Huiping Cao, Nikos Mamoulis, and David W., "Mining Frequent Spatio-temporal Sequential Patterns", *Department of Computer Science The University of Hong Kong Pokfulam Road, Hong Kong* {*hpcao, nikos, dcheung*}*@cs.hku.hk.*

[3] Mikołaj Morzy, "Mining Frequent Trajectories of Moving Objects for Location Prediction", Institute of Computing Science Pozna´n University of Technology
Piotrowo 2, 60-965 Pozna´n, Poland Mikolaj.Morzy@put.poznan.pl.

[4] Torben Bach Pedersen Aalborg University tbp@cs.aau.dk, Manohar Kaul Uppsala University manukaul@acm.org,
Gy˝oz˝o Gidófalvi KTH Royal Inst. of Technology gyozo.gidofalvi@abe.kth.se, Christian Borgelt EU Centre for Soft Computing christian@borgelt.net, "Frequent Route Based Continuous Moving Object Location- and Density Prediction on Road Networks" .

[5] Yunhao Liu, Yiyang Zhao, Lei Chen, Jian Pei, Jinsong Han, "Mining Frequent Trajectory Patterns for Activity Monitoring Using Radio Frequency Tag Arrays", IEEE.