Journal of Zhejiang University SCIENCE A
ISSN 1673-565X (Print); ISSN 1862-1775 (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

Personal continuous route pattern mining*

Qian YE, Ling CHEN‡, Gen-cai CHEN
(*School of Computer Science and Technology, Zhejiang University, Hangzhou 310027, China*)
E-mail: yeqian.zju@gmail.com; lingchen@zju.edu.cn; chengc@zju.edu.cn

Received Mar. 17, 2008; Revision accepted June 10, 2008; Crosschecked Dec. 25, 2008

Abstract: In the daily life, people often repeat regular routes in certain periods. In this paper, a mining system is developed to find the continuous route patterns of personal past trips. In order to count the diversity of personal moving status, the mining system employs the adaptive GPS data recording and five data filters to guarantee the clean trips data. The mining system uses a client/server architecture to protect personal privacy and to reduce the computational load. The server conducts the main mining procedure but with insufficient information to recover real personal routes. In order to improve the scalability of sequential pattern mining, a novel pattern mining algorithm, continuous route pattern mining (CRPM), is proposed. This algorithm can tolerate the different disturbances in real routes and extract the frequent patterns. Experimental results based on nine persons' trips show that CRPM can extract more than two times longer route patterns than the traditional route pattern mining algorithms.

Key words: Data mining, Route pattern, GPS, Mobile phone
doi:10.1631/jzus.A0820193 **Document code:** A **CLC number:** TP39

INTRODUCTION

With the prevalence of portable devices [e.g., mobile phones and personal digital assistants (PDAs)] and positioning devices (e.g., GPS), recording the trips of moving objects and mining their route patterns become possible. Based on these technologies, the quality of services of many systems can be improved. For example, the route patterns of vehicles collected by a traffic scheduling system can be used to estimate traffic conditions. The route patterns of mobile devices can be utilized by the providers of location-based services (LBSs) to distribute resources. Moreover, the route patterns of moving objects can also be used to predict the future routes. Route prediction has many potential applications. For example, if vehicle navigation systems knew the future routes, they can provide real-time traffic notifications to drivers. Researchers from Nissan showed that it is possible to improve hybrid fuel economy by up to 7.8% if the routes of vehicles are known in advance (Deguchi *et al.*, 2003).

These promising applications have inspired the research on route pattern mining. However, most existing work mainly focuses on the route patterns of vehicles but not the persons, while the latter is more universal. The routes of vehicles are much less than the personal routes since most people only take vehicles for their long-distance travels.

In this paper, we focus on the personal route pattern mining. Different from the route pattern mining for vehicles, there are three unique challenges with personal route pattern mining. First, personal moving status is more diverse than that of a vehicle. The velocities of vehicles vary in a relatively small range because of the speed limit. However, a person's moving speed may change in a larger range, depending on his/her traveling methods. Moreover, many people spend most of their time in buildings where the GPS devices suffer from drifting signals, while vehicles are often outside buildings. Second, personal route patterns involve vital information that should be kept away from potential risks, because personal

‡ Corresponding author

* Project supported by the National Natural Science Foundation of China (No. 60703040) and the Natural Science Foundation of Zhejiang Province, China (No. Y107178)

routes show every place a person has been to. It implicates that some mining procedures should be done on personal mobile devices. Another accompanying problem is the limited computational capability of mobile devices. Unlike the computational devices carried by vehicles, the computational capability of mobile devices for persons is restricted by their volume and battery power.

We build a personal continuous route pattern mining system to tackle these challenges. In our work, GPS devices are used to obtain users' trips data. The recording program can adaptively adjust the sampling interval according to moving status. Five data filters are designed to remove the outliers from source data. Adaptive recording and data filters make the system be suitable for the diversity of personal moving status. The client/server architecture of the system guarantees the security of users' privacy, and reduces the computational load on mobile devices. Moreover, a novel mining algorithm, continuous route pattern mining (CRPM), based on PrefixSpan (Pei *et al*., 2001), is proposed to extract continuous route patterns with improved scalability. An incremental mining strategy is also available to support the system for long-term running.

The rest of the paper is organized as follows. Section 2 gives an overview of the related work. Section 3 describes the details of our system in terms of route segmentation, data cleaning, and mining. Section 4 contains experimental results and discussion. Finally, Section 5 concludes the paper and gives potential future work.

BACKGROUND

So far, the research on the route patterns of moving objects can be classified into two categories according to the mining process. In the first one, the route patterns are built directly based on the real observation data. For example, Froehlich and Krumm (2008) used a modified version of the Hausdorff distance to merge similar trips of vehicles for obtaining the regular routes. The computational complexity of this kind of method is higher than the other mining strategy, more popular in this research field, which applies some simplification technologies to abstract the real observation data and conducts mining on the simplified trips. Among the examples of the second method, Simmons *et al*.(2006) made use of lines in road maps to represent corresponding real routes data and trained a hidden Markov model to describe the route patterns of vehicles; Cao *et al*.(2005) used the lines simplification method to express line trips by rectangles and utilized a substring tree for mining; Giannotti *et al*.(2006) utilized sequences of cells to simplify the real trips and designed a mining algorithm called MiSTA based on PrefixSpan (Pei *et al.*, 2001) and took time spans into consideration, and then Giannotti *et al.*(2007) used MiSTA in trajectory pattern mining.

However, all these works face two common challenges. First, they have not proposed suitable methods to deal with the diversity of personal moving status. They all focus on the route patterns of vehicles. However, the moving status of vehicles is different from that of persons, as discussed in Section 1. Second, they have not considered the personal privacy. Actually, the privacy issue in trajectory mining has been paid particular attention to. For example, Gedik and Liu (2008) proposed an architecture, which used a flexible privacy personalization framework to support location *k*-anonymity for a wide range of mobile clients with context-sensitive privacy requirement. The GeoPKDD project (http://www.geopkdd.eu/) takes the privacy issue as one of its major research topics. In this project, Abul *et al*.(2007) studied the privacy preserving data mining methods for moving object data, and devised a heuristics and sanitization algorithm. Abul *et al*.(2008) then conducted the research on (k, δ)-anonymity for moving objects databases, and proposed a greedy algorithm based on clustering. These works proposed to protect personal privacy by hiding the identifiable precise position of a user from the public. Laasonen (2005) also discussed the privacy issue. He exploited a GSM network to retrieve the location information, and ran a pattern mining procedure on the mobile phone. His work inspired another way to protect personal privacy, that is, to perform some mining procedures on personal mobile devices. In the proposed mining system, we guarantee the users' privacy by saving the geographic information in the personal mobile devices. The data sent to the server only contain sequences without any geographic information. It is impossible to recover the real trip data from these sequences.

MINING CONTINUOUS ROUTES

Our mining system includes two separated parts: the client part and the server part. Fig.1 shows the data flows of the mining system.

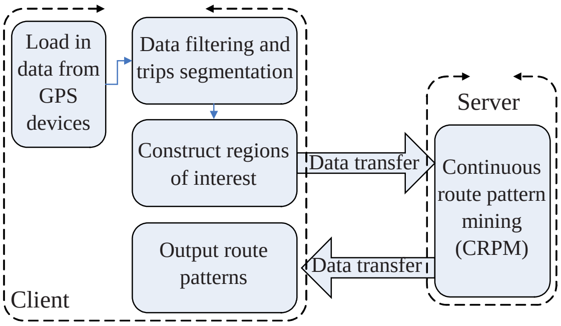

Load in data from GPS devices
Data filtering and trips segmentation
Construct regions of interest
Output route patterns
Client
Data transfer
Data transfer
Server
Continuous route pattern mining (CRPM)

Fig.1 Data flows of the mining system

As shown in Fig.1, the data collecting, data filtering, and construction of interested regions are done at the client part. The server only gets regional-temporal sequences (RTSs) and is responsible for CRPM. This architecture makes the server take most of the computation while keeping users' real routes at the client part, and thus keeps the privacy of users.

This section contains the detailed four major parts of the mining system: data collecting, trip filtering, pattern mining, and incremental mining.

Data collecting

In data collecting, the personal route recording is different from that of vehicles. As mentioned in Section 1, the moving status of a person is much more diverse than that of a vehicle. The person may take different transportation at different speeds, or stay in a building for a long time, where GPS devices suffer from drifting signals. It is the reason that the positions have to be recorded at different rates according to moving status, and clear outlier positions.

In our work, mobile phones are chosen as the recording devices. A program written in Python running on the mobile phone can be connected to a GPS device through Bluetooth and record the GPS positions with time stamps. What the users need to do is turning on their mobile phones and GPS devices in the morning, bringing these mobile devices with them, and turning off them at night. The recording system can recover automatically from a Bluetooth failure and adaptively adjust the sampling interval according to the moving speed of the GPS device. This feature helps to record personal trips under diverse moving status. Fig.2 shows the finite state automaton of the recording program.

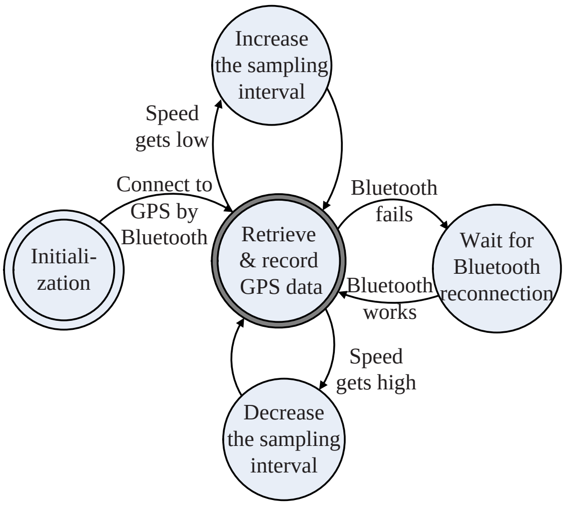

Increase the sampling interval
Speed gets low
Connect to GPS by Bluetooth
Initialization
Retrieve & record GPS data
Bluetooth fails
Wait for Bluetooth reconnection
Bluetooth works
Speed gets high
Decrease the sampling interval

Fig.2 Finite state automaton of the recording program

Trip filtering

Due to the uncertainty of GPS, the raw data recorded by mobile phones contain many outliers (Fig.3) (Throughout the paper, all the trips are displayed in Google Earth^{TM}), which need to be removed. In this step, the difference for the work on personal routes is that it is hard to segment the personal GPS data into independent trips. GPS devices installed in a vehicle can segment trips based on the status of the engine. They can begin recording GPS data when the engine starts and pause when the engine stops (Froehlich and Krumm, 2008). However, asking users to manually turn on/off their devices several times a day would decrease the usability of the system. Thus, five filters are developed to remove these outliers and segment trips according to different criteria.

(1) Duplication filter: The duplication filter removes the second position if the distance between two consecutive positions in the raw data is smaller than a threshold λ_{dup}.

(2) Speed filter: Assuming that individuals move at a constant speed between two consecutive positions, and that there is a reasonable speed range for individuals, the speed filter removes the second position if the speed between two consecutive positions is unreasonable.

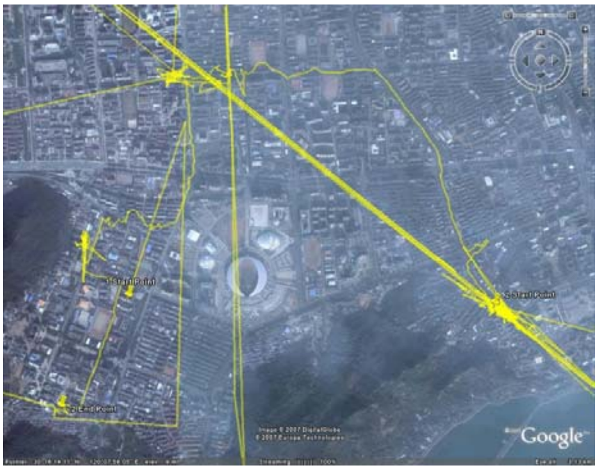


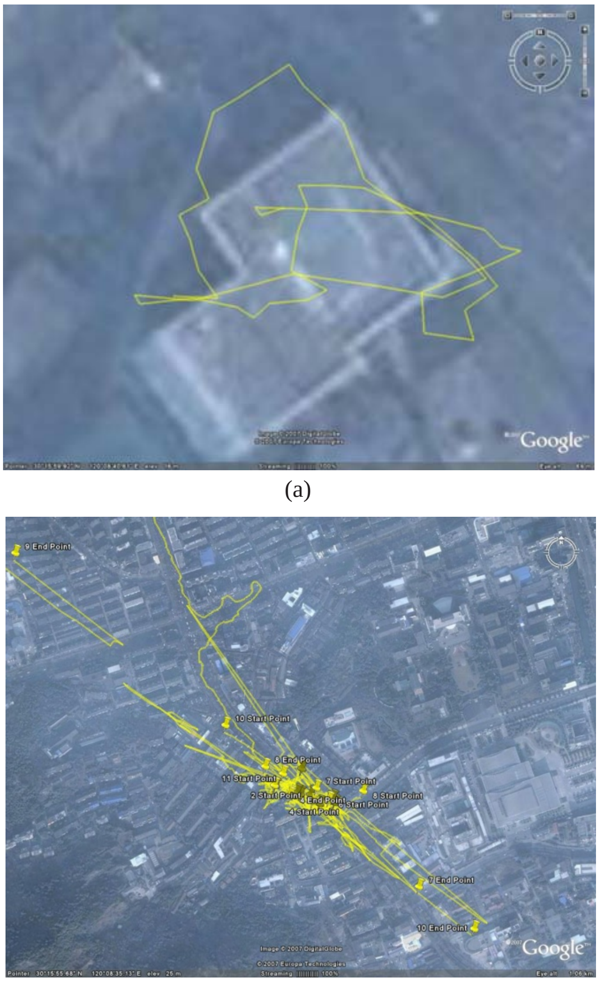
Fig.3 Raw routes data recorded by mobile phones

(3) Acceleration filter: The acceleration filter can remove the positions that contribute to an acceleration of over a threshold.

(4) Total-distance filter: The total-distance filter is designed to remove the redundant GPS data recorded when users are inside buildings. Given a window size δ, first the centroid of each δ sequential positions in a trip is calculated. Then the maximum distance between these centroids is determined to estimate whether the trip contributes to a reasonable moving distance. A route will be dropped, if its maximum centroid is less than a threshold λ_{tdis}. Fig.4a gives an example of useless trips that can be dropped by the total-distance filter.

(5) Angle filter: The angle filter is used to smooth the routes. By reading the three sequential positions A, B, C, it calculates the angle of $\angle ABC$. Because the sampling interval between two conterminous positions is small (less than 5 s in a particular implementation), if $\angle ABC$ is smaller than a threshold λ_{ang}, position B is probably an outlier, since people are unlikely to take sharp turns during a few seconds. The angle filter is scheduled to run repeatedly for each trip until there is no outlier to remove. In Fig.4b, many outliers in trips belong to this kind, and can be cleared by the angle filter.

The recorded trips are segmented with these filters. The basic criterion for splitting GPS data is the time gap between two consecutive positions, since a moving stop indicates the end of a trip. Algorithm 1 is used to segment trips using data filters, where T is the array containing all recorded trips of a person, $\lambda_{\text{time\_gap}}$ is the time threshold used to segment trips, $\lambda_{\text{trj\_cnt}}$ is the threshold used to remove short trips, and *Funct*() is one of the data filtering functions described above. *Funct*() returns 'true' if the positions comply with the restriction of the data filter; otherwise, it returns 'false', and the corresponding positions will be removed.



(a)

(b)

Fig.4 Examples of the noisy GPS raw data
(a) Useless trips that can be dropped by the total-distance filter; (b) Outliers in trips that can be cleared by the angle filter

Algorithm 1 Segment trips using data filters

```
Input: $T$, $\lambda_{\text{time\_gap}}$, $\lambda_{\text{trj\_cnt}}$, $Funct()$
Output: $T_{\text{tmp}}$
$T_{\text{tmp}}=\varnothing$;
for (each route $r_i$ in $T$) do
    $r_{\text{tmp}}=\varnothing$;
    for (each position $p_j$ in $r_i$) do
        if ($Funct(p_j, r_i)$ returns true)
            $r_{\text{tmp}}=Append(r_{\text{tmp}}, p_j)$;
        else if ($Time(p_j)-Time(p_{j-1})>\lambda_{\text{time\_gap}}$)
            if ($Size(r_{\text{tmp}})>\lambda_{\text{trj\_cnt}}$)
```

```
9.              $T_{\text{tmp}}$=Append($T_{\text{tmp}}$, $r_{\text{tmp}}$);
10.           endif
11.       endif
12.   endfor
13. endfor
14. return $T_{\text{tmp}}$;
```

The data filtering process can remove the noisy raw data (Fig.5), and greatly reduce the amount of the original real trip data. The output trips are organized as spatio-temporal sequences (STSs), in the form of $\langle(x_0,y_0,t_0), (x_1,y_1,t_1), \ldots, (x_k,y_k,t_k)\rangle$, where (x_i,y_i) (i=0, 1, …, k) is a longitude-latitude pair and t_i (i=0, 1, …, k) is a time stamp.

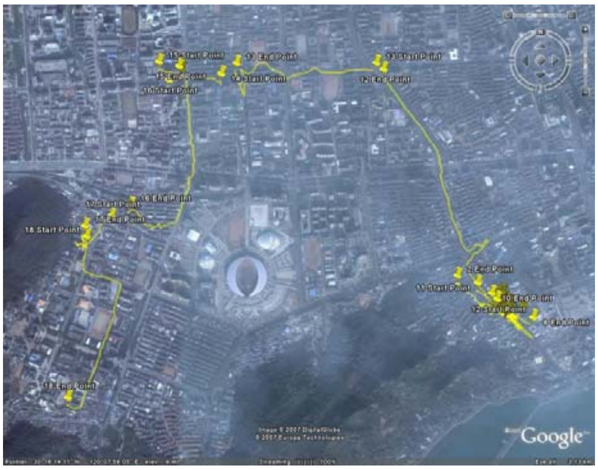


Fig.5 The same routes as in Fig.3 after the filtering stage

Route pattern mining

The proposed CRPM algorithm is based on the PrefixSpan algorithm (Pei *et al.*, 2001), and can improve the scalability of the pattern mining. Algorithm 2 shows the main algorithm for route pattern mining.

Algorithm 2 *CRPM*(*STS*, λ_{time})

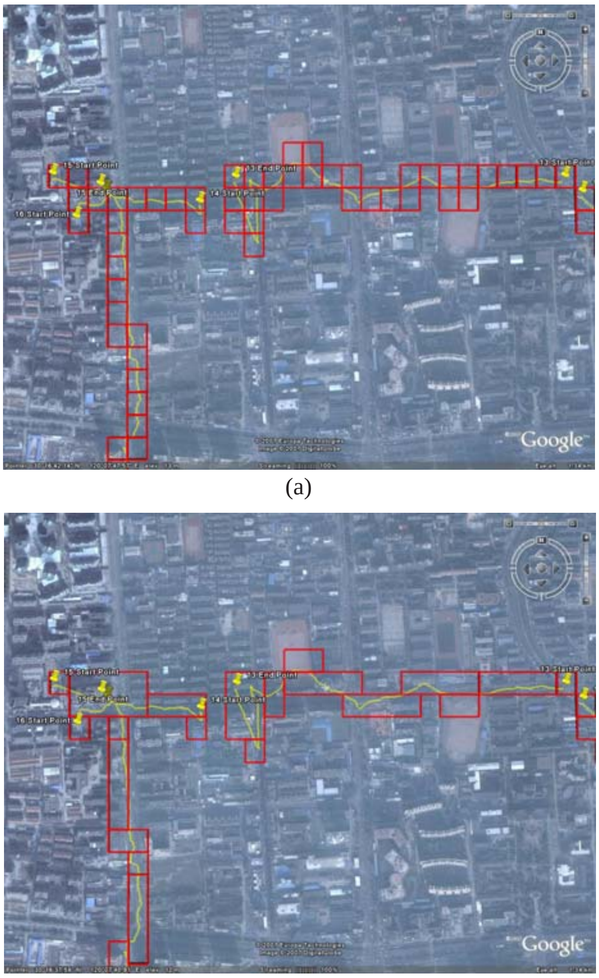
```
Input: STS, $\lambda_{\text{time}}$
Output: PS
1.  grid=build_grid(STS);
2.  CTS=interpolation(grid, STS);
3.  for (each item $GTelem_i$ in CTS) do
4.      compute_density(grid, $GTelem_i$);
5.  endfor
6.  if (incremental mining) do
7.      load_past_trajctories(STS, dbFile);
8.      update_density(grid, gridFile);
9.  endif
10. Regions=bound_regions_of_interest(grid);
11. RTS=translation(CTS, Regions);
12. PS={RTS};
13. while (PS is not empty) do
14.     P=pop(PS);
15.     PS′=extend_projection(P, $\lambda_{\text{time}}$);
16.     PS=Append(PS, PS′);
17. endwhile
```

Our mining approach starts by using sequences of cells to simulate the real personal trips (line 1 in Algorithm 2). Given the STSs of a person, the algorithm equally divides his/her active area (the area one has visited) into grids. So his/her real trips can be represented in sequences of cells. Linear interpolation is used to make sure that all the cells that users have passed can be extracted (line 2 in Algorithm 2). In Algorithm 2, the cell-temporal sequence (CTS) is in the form $\langle(\tilde{C}_0,t_0), (\tilde{C}_1,t_1), ..., (\tilde{C}_k,t_k)\rangle$, where $\tilde{C}_i$ (i=0, 1, ..., k) is a cell, and t_i (i=0, 1, ..., k) is a time stamp.

The method to construct the regions that are frequently visited (regions of interest, ROIs) is based on the visiting density (lines 3~10 in Algorithm 2), which is similar to the work of (Giannotti *et al*., 2007). A threshold is employed to judge whether a cell is frequently accessed. The concatenate cells with similar densities are merged to regions. The next step is to transform STSs to RTSs based on the ROIs (line 11 in Algorithm 2).

Definition 1 (Regional-temporal sequence, RTS) An RTS is a sequence of couples $\bar{S}=\langle(\bar{R}_0,\bar{T}_0), (\bar{R}_1,\bar{T}_1), ..., (\bar{R}_k,\bar{T}_k)\rangle$, in which region $\bar{R}_i$ (i=0, 1, …, k) is a set of cells, and $\bar{T}_i=(T_{\text{in}}^{(i)},T_{\text{out}}^{(i)})$ $(i=0,\ 1,\ ...,\ k)$. $\forall 0\le i<k$, $i\in\mathbb{N}^*$, $T_{\text{in}}^{(i)}<T_{\text{out}}^{(i)}\le T_{\text{in}}^{(i+1)}$.

In Definition 1, $\bar{R}_i$ is one of the ROIs constructed at the previous step. T_{in} is the time when the person enters this region, while T_{out} is the leaving time. According to the definition, given two items S_m, S_n (m, n=0, 1, …, k) in an RTS, there is no time overlap between these two items, although R_m, R_n may represent the same region. Moreover, $T_{\text{out}}^{(m)}$=$T_{\text{in}}^{(n)}$ means that item n is the sequential one right after item m. In the system, the entering time is set as the time stamp of the first position in the region, while the leaving time is set as the time stamp of the last one. Figs.6a and 6b show an example trip represented in CTS and RTS, respectively.



(a)

(b)

Fig.6 The trips displayed in cell-temporal sequences (a) and regional-temporal sequences (b)

All the mining preprocesses described above are performed on users' mobile devices. After that, the RTSs are transferred to the server. Since the system only sends the IDs of regions in RTSs to the server, the information received by the server is insufficient to recover the real trips of users, and thus the privacy of users is protected. The mining step on the server is the key step of our algorithm. We first define the continuous route as follows:

Definition 2 (Continuous route) An RTS represents a continuous route iff $\forall 0 \le i < k,\ i \in \mathbb{N}^*, T_{\text{in}}^{(i+1)} - T_{\text{out}}^{(i)} \le \lambda_{\text{time}}$.

In Definition 2, λ_{time} is the maximum time gap between two consecutive areas in a route pattern. When λ_{time}=0, the mining results equal the results of a substring mining algorithm, e.g., substring tree mining (Cao *et al.*, 2005). When $\lambda_{\text{time}} \to +\infty$, the mining results equal the results of a typical sequential pattern mining algorithm, e.g., PrefixSpan (Pei *et al.*, 2001).

The advantage of the proposed mining algorithm is that it can tolerate the diversity in trips and reserve the continuous properties of trips in patterns. Fig.7 shows an example. There are three trips starting from the top-left corner and leading to the bottom-right corner. Given the minimum support threshold $\lambda_{\text{min\_sup}}$=3, the longest sequential pattern a substring mining algorithm can find is {B, C, D, E, F}. However, the three trips actually start from the same building, and they turn to different directions because there is a parterre in front of the building. If the time spent on passing the parterre is considered as an acceptable time gap, a longer route pattern, {A, B, C, D, E, F}, can be found. The blank between A and B will not affect the continuity of this route pattern, because the distance is short in the real world and there is no fork. Actually, the parterre can be any other disturbance in daily life, like obstacles, drifting signals, and short-term device unavailability. In this way, CRPM can improve the scalability of the mining algorithm.

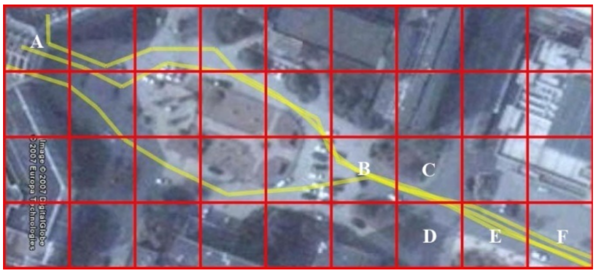
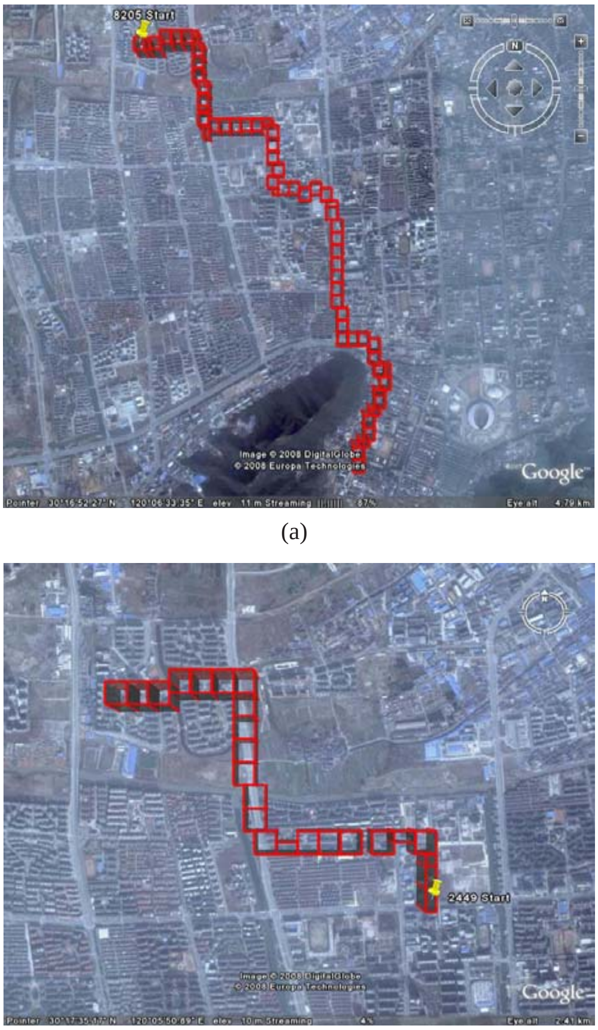


Fig.7 An example of a user's trips

The CRPM algorithm is different from the PrefixSpan algorithm for introducing a threshold λ_{time}. The critical function (line 15 in Algorithm 2) is described in Algorithm 3. In this function, *P* is a projection that contains a continuous prefix and the RTSs that contain the prefix. *P* will be extended to a new regional element if the time gap between the last element of the prefix and the new element is less than the threshold λ_{time}. The continuous route patterns are recorded in RTSs and transferred back to the client for further usages. Fig.8 shows some examples of extracted route patterns.

Algorithm 3 *extend_projection*(P, λ_{time})

```
Input: $P$, $\lambda_{\text{time}}$
Output: $P'$
$PS=\varnothing$;
for (each $RTS_i$ in $P$) do
    $lastRTelem=get\_lastProj(RTS_i)$;
    for (each item $RTelem_j$ in $RTS_i$) do
        if ($RTelem_j.in-lastRTelem.out<\lambda_{\text{time}}$)
            $P'=generate\_proj(RTS_i, RTelem_j)$;
        else
            break;
        endif
        $update\_support(Append(P.prefix, RTelem_j.R), P')$;
        $PS=Append(PS, P')$;
    endfor
endfor
return $P'$;
```


8205 Start
Image © 2008 DigitalGlobe
© 2008 Europa Technologies
Google
2449 Start
Image © 2008 DigitalGlobe
© 2008 Europa Technologies
Google

(a)

(b)

Fig.8 Two examples of continuous route patterns
(a) The road pattern of a faculty from home to the campus; (b) The road pattern of a student from the campus to home

Incremental mining

Two strategies are utilized in the proposed system to reduce the computational load on mobile devices. First, the client/server architecture makes the main mining procedure run on the server. Second, the incremental mining is implemented in the system.

On the client side, the most time-consuming function is the computation of density (lines 3 and 4 in Algorithm 2), whose computational complexity is $O(N^2)$. The incremental mining strategy attempts to reuse the results of previous mining, i.e., recovering the past CTSs and densities of cells. As a result, the incremental mining can greatly reduce the computational load while guaranteeing the correctness of the mining results. Lines 7 and 8 in Algorithm 2 show the incremental mining strategy. A corresponding restriction of incremental mining is that the grid cannot be resized.

RESULTS AND DISCUSSION

In our experiments, the mobile phone Nokia N70 was used as the recording device, and the GPS sensor HOLUX 1000 was used to obtain GPS data. The recording program running on Nokia N70 was written based on Pysumbler (http://www.mrl.nott.ac.uk/~lxo/ipergtools/01_pystumbler.htm). Eleven participants, who are all students and faculties of Zhejiang University, were involved in the experiments. They lived at different places in the same city of Hangzhou and each has his/her own regular trips. Every participant took part in the experiment for more than one month. They carried the experimental devices with them all through the days. Table 1 shows the total of the recorded positions of the participants.

Table 1 The total of recorded positions for every participant

Participant ID	Total recorded positions	Time period (MM.DD.YYYY)
1	335 664	11.12.2007~12.20.2007
2	196 583	12.23.2007~01.25.2008
3	43 700	11.08.2007~12.22.2007
4	159 880	11.08.2007~12.24.2007
5	52 541	12.24.2007~01.25.2008
6	130 932	11.13.2007~12.24.2007
7	151 961	12.26.2007~01.25.2008
8	17 616	01.17.2008~02.16.2008
9	95 103	12.24.2007~01.25.2008
10	133 117	01.25.2008~02.20.2008
11	175 520	10.10.2007~02.20.2008

The total number of recorded positions varies in a large range for several practical reasons. First, participants used different traveling methods, which resulted in different sampling rates. For example, Participant 6 often took a bus when he went from the campus to the company for his part-time job, while Participant 9 often rode bicycle. Second, participants had different trips and spent different time on traveling. Third, the participants may forget to turn on their devices.

In the data filtering procedure, the data filters were applied to the GPS data in this order: duplication filter, speed filter, acceleration filter, total-distance filter, and angle filter. It is apparent that these data filters are not independent. An outlier to one filter may be also an outlier to another filter. For example, the outliers that contribute to high speed may also indicate high acceleration. So it is hard to evaluate the efficiency of each data filter. Another challenge is that the parameters of these filters are hard to tune. Due to the precision limitation of the GPS device, there is no ground truth for source data to classify all the outliers correctly. In our work, the parameters were set by data analysis and experience.

We set the time threshold $\lambda_{\text{time\_gap}}$ to 120 s to tolerate some random interruption during traveling. For instance, the red traffic light will stop vehicles, and meeting friends on the street will also make one pause. The speed threshold and acceleration threshold were set to 27 m/s and 10 m/s^2, respectively. The λ_{ang} was set to π/6. Fig.9 is the total positions of each participant before and after the filtering. It shows that about 75% original positions are filtered as redundant positions or outliers by the filters. Fig.10 shows the total number of trips for each participant. As mentioned in the subsection of “Trip filtering”, the trips that contain less than 20 positions will be dropped by the filters. Trips data from Participants 3 and 8 were thus dropped, and we took the GPS data of the remaining nine participants as the original data for our mining experiments.

The performance of the route pattern mining can be adjusted through several parameters, which provides more scalability in mining. The parameter *max_reg_size* sets the maximum total number of cells on a side of an ROI. In our experiment, the default side length of a cell is 50 m, which means the parameter *max_reg_size* restricts the maximum side length of an ROI to *max_reg_size*×50 m. This parameter represents a tradeoff between the precision and computational load. A small *max_reg_size* provides more accurate mining results while suffers more computational cost, because there will be more ROIs. Fig.11 discloses the relation between *max_reg_size* and the total number of ROIs for each participant. We set $\lambda_{\text{min\_sup}}$ to 5 in our experiments.

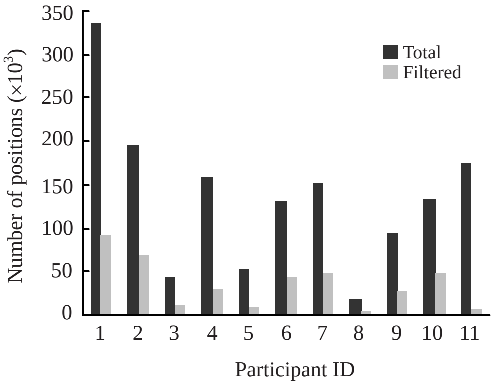


Fig.9 The total of positions of each participant before and after filtering

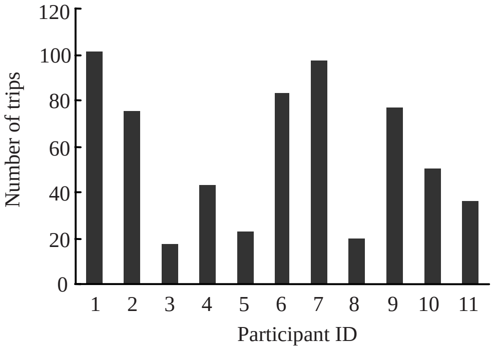


Fig.10 The total of trips of the participants

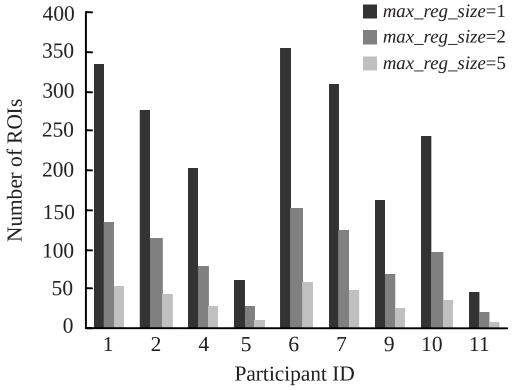


Fig.11 The total number of regions of interest (ROIs) for the remaining 9 participants vs. different *max_reg_size*

Participants 3 and 8 are dropped as their trips contain less than 20 positions, as shown in Fig.10

The key parameter for CRPM is λ_{time}, which defines the maximum time interval that can be accepted in a continuous route, and $\lambda_{\text{time}}<\lambda_{\text{time\_gap}}$. As discussed in the subsection of “Route pattern mining”, when $\lambda_{\text{time}}=0$, CRPM equals the substring mining [e.g., substring tree mining in (Cao *et al*., 2005)]. According to the definition of CRPM, along with the increase of λ_{time}, the total number of route patterns (including the route patterns of all possible lengths) will increase. Fig.12 shows this trend when increasing λ_{time}. Because the absolute value of the total number of route patterns is different from one participant to another, we express the proportional relationships in Fig.12. The total number of route patterns when $\lambda_{\text{time}}=0$ is the baseline for each participant. The results of the comparison between the traditional sequential mining and CRPM are shown in the figure. The *max_reg_size* was set to 2 in this experiment. We can find in Fig.12 that, when $\lambda_{\text{time}}=10$ the total number of patterns of some participants (e.g., Participants 1, 2 and 10) increase much more than those of other participants. It is because these participants have longer regular trips by taking fast vehicles. Since more random interruption is tolerated by λ_{time}, more new route patterns and sub-patterns are extracted.

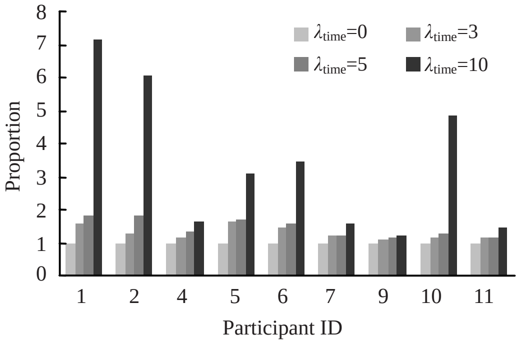


Fig.12 The proportional relationships between the totals of patterns when λ_{time}=0, 3, 5, and 10
The total number of route patterns when $\lambda_{\text{time}}=0$ is the baseline for each participant

The most important benefit of CRPM is that it can extract longer continuous route patterns than the ordinary substring mining. To demonstrate it, we define “maximum distinct subset (MDS)” as follows.

Definition 3 (Regional sequence containment, $\supseteq$) A regional sequence $\overline{S}_1$ $(=\langle \overline{R}_0, \overline{R}_1, ..., \overline{R}_j\rangle)$ contains another regional sequence $\overline{S}_2$ $(=\langle \overline{R}_0, \overline{R}_1, ..., \overline{R}_k\rangle)$, i.e., $\overline{S}_1 \supseteq \overline{S}_2$, iff $\overline{S}_2$ is a subsequence of $\overline{S}_1$.

Definition 4 (Maximum distinct subset, MDS) A regional sequences set $\overline{P}_1$ is an MDS of another regional sequences set $\overline{P}_2$ iff $\forall \overline{S}_2 \in \overline{P}_2$, $\exists \overline{S}_1 \in \overline{P}_1 \wedge \overline{S}_1 \supseteq \overline{S}_2$ and $\forall \overline{S}_1 \in \overline{P}_1$, $\nexists \overline{S}_3 \in \overline{P}_1 \wedge \overline{S}_1 \supseteq \overline{S}_3 \wedge \overline{S}_1 \neq \overline{S}_3$.

It is shown that MDS is a set without redundant route patterns, which can properly demonstrate the advantage of CRPM. Table 2 shows the longest and average length of route patterns in MDS for different λ_{time}, for each participant. Notice that when $\lambda_{\text{time}}=0$, CRPM equals the traditional substring mining. Table 2 clearly shows that increasing λ_{time} helps to find longer route patterns.

Figs.13a and 13b are examples of Participants 1 and 5, respectively, which present how λ_{time} affects the route patterns of different length in MDSs. It is shown that, with the increase of λ_{time}, the total numbers of long route patterns increase fast. Especially, the total of route patterns in Fig.13a shows a great increase when $\lambda_{\text{time}}=10$. It is because Participant 1 has longer regular daily trips. When random interruption is covered by λ_{time}, the number of route patterns with middle length grows quickly.

Table 2 The longest and average length of route patterns in MDS for each participant

Participant ID	Longest length (regions)				Average length (regions)			
	$\lambda_{\text{time}}=0$	3	5	10	$\lambda_{\text{time}}=0$	3	5	10
1	27	32	33	42	8.691	11.619	12.291	19.238
2	29	34	35	42	10.648	12.054	14.873	20.371
4	11	13	15	17	4.566	4.963	5.577	6.482
5	12	18	18	18	5.604	7.789	8.140	9.971
6	31	45	45	45	7.072	9.240	10.506	15.099
7	7	10	10	12	2.481	2.756	2.765	3.214
9	10	11	11	11	2.763	2.893	2.921	3.060
10	24	25	25	37	8.292	9.497	9.915	16.133
11	4	5	5	7	2.000	2.179	2.214	2.690

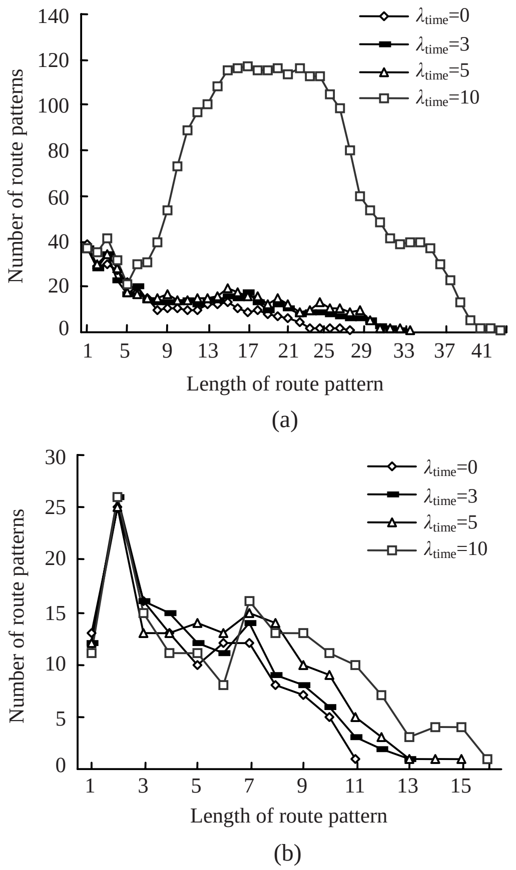


Fig.13 The total of route patterns of different lengths for Participant 1 (a) and Participant 5 (b)

We also conducted human judgment to evaluate the performance of our mining system. After the experiment, the nine participants, whose route data were used in our experiments, were asked to answer a questionnaire. There are five randomly selected route patterns from their own 50% longest route patterns.

For each of their route patterns, the nine participants are asked the following three questions:

Question 1: Is it a route pattern for you?

Question 2: What's your traveling frequency on the route?

Question 3: What's the proportion of the route pattern to your regular trip, containing the route, in length?

The results show that all the participants consider the extracted route patterns as their route patterns. Fig.14 is the results of the second question, which shows that most of the extracted route patterns are traveled more than once a week. It is partly because the participants took part in the data recording procedure for only about one month. The recorded data are insufficient to support the mining route patterns of lower frequencies. Fig.15 shows the results of the third question, which is a kind of precision evaluation for our mining system. It shows that most of route patterns are considered to cover more than 50% of the real regular trips. However, there are still some short patterns due to several practical reasons, for example, frequent GPS signal drifting, battery failure. This problem can be overcome by using more source data.

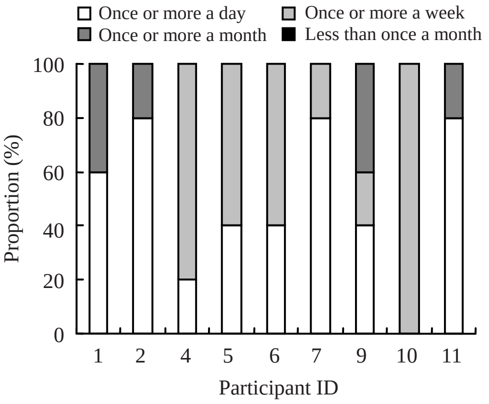


Fig.14 The frequency of the route patterns for each participant

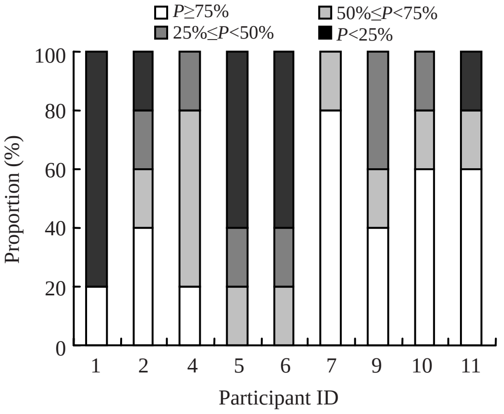


Fig.15 The proportion of route patterns to real trips for each participant in length

P refers to the proportion of the route pattern to the real regular trip

CONCLUSION

This paper shows that the personal route patterns can be extracted using a mobile phone, a low-cost GPS sensor and a common computer. We proposed a practical personal route pattern mining system that locates the position by a GPS sensor and considers diverse information including individual moving status, personal privacy, and scalability of route patterns. The incremental mining strategy supports the mining system to reuse the previous results in future mining work. The experiments on nine participants show that the system can extract more and longer route patterns than the route pattern mining methods based on the traditional sequential mining methods.

In the future, our work can be extended in two aspects. First, more additional information can be utilized for better mining results. For example, the road information can be used to resize the grid; the time information can be utilized to build different route patterns for different time periods. Second, the route patterns can be used to predict personal future routes, which can help to improve the quality of personal location-based services (LBSs).

References

Abul, O., Atzori, M., Bonchi, F., Giannotti, F., 2007. Hiding Sensitive Trajectory Patterns. Seventh IEEE Int. Conf. on Data Mining Workshops, p.693-698. [doi:10.1109/ICDMW.2007.93]

Abul, O., Bonchi, F., Nanni, M., 2008. Never Walk Alone: Uncertainty for Anonymity in Moving Objects Databases. Proc. Int. Conf. on Data Engineering, p.376-385. [doi:10.1109/ICDE.2008.4497446]

Cao, H.P., Mamoulis, N., Cheung, W.D., 2005. Mining Frequent Spatio-Temporal Sequential Patterns. Proc. Int. Conf. on Data Mining, p.82-89. [doi:10.1109/ICDM.2005.95]

Deguchi, Y., Kuroda, K., Shoji, M., 2003. HEV charge/discharge control system based on navigation information. *Proc. JSAE Ann. Congr.*, **29**(3):1-4.

Froehlich, J., Krumm, J., 2008. Route Prediction from Trip Observations. Proc. Intelligent Vehicle Initiative (IVI) Technology Advanced Controls and Navigation System. SAE World Congress & Exhibition.

Gedik, B., Liu, L., 2008. Protecting location privacy with personalized *k*-anonymity: architecture and algorithms. *IEEE Trans. Mobile Comput.*, **7**(1):1-18. [doi:10.1109/TMC.2007.1062]

Giannotti, F., Nanni, M., Pedreschi, D., 2006. Efficient Mining of Temporally Annotated Sequences. Proc. SIAM Int. Conf. on Data Mining, p.346-357.

Giannotti, F., Nanni, M., Pedreschi, D., Pinelli, F., 2007. Trajectory Pattern Mining. Proc. Int. Conf. on Knowledge Discovery and Data Mining, p.330-339. [doi:10.1145/1281192.1281230]

Laasonen, K., 2005. Clustering and Prediction of Mobile User Routes from Cellular Data. Proc. European Conf. on Principles of Data Mining and Knowledge Discovery, p.569-576.

Pei, J., Han, J., Mortazaviasl, B., Pinto, H., Chen, Q., Dayal, U., Hsu, M., 2001. PrefixSpan: Mining Sequential Patterns by Prefix-Projected Growth. Proc. Int. Conf. on Data Engineering, p.215-224.

Simmons, R., Browning, B., Zhang, Y., Sadekar, V., 2006. Learning to Predict Driver Route and Destination Intent. IEEE Intelligent Transportation Systems Conf., p.127-132. [doi:10.1109/ITSC.2006.1706730]