

# Mining Movement Data to Extract Personal Points of Interest: A Feature Based Approach

Marco Pavan, Stefano Mizzaro and Ivan Scagnetto

**Abstract** Due to the widespread of mobile devices in recent years, records of the locations visited by users are common and growing, and the availability of such large amounts of spatio-temporal data opens new challenges to automatically discover valuable knowledge. One aspect that is being studied is the identification of important locations, i.e. places where people spend a fair amount of time during their daily activities; we address it with a novel approach. Our proposed method is organised in two phases: first, a set of candidate stay points is identified by exploiting some state-of-the-art algorithms to filter the GPS-logs; then, the candidate stay points are mapped onto a feature space having as dimensions the area underlying the stay point, its intensity (e.g. the time spent in a location) and its frequency (e.g. the number of total visits). We conjecture that the feature space allows to model aspects/measures that are more semantically related to users and better suited to reason about their similarities and differences than simpler physical measures (e.g. latitude, longitude, and timestamp). An experimental evaluation on the GeoLife public dataset confirms the effectiveness of our approach and sheds some light on the peculiar features and critical issues of location based systems.

## 1 Introduction

In recent years, the increasing pervasiveness of mobile devices and the ever growing mobile technologies have made location-acquisition systems available to everyone. Moreover, such systems can be easily embedded in popular apps and services, being very often active during many users' daily activities. This evolution allows to collect

---

M. Pavan (✉) · S. Mizzaro · I. Scagnetto  
Department of Mathematics and Computer Science, University of Udine,  
Via Delle Scienze 206, Udine, Italy  
e-mail: marco.pavan@uniud.it

S. Mizzaro  
e-mail: mizzaro@uniud.it

I. Scagnetto  
e-mail: ivan.scagnetto@uniud.it

large datasets with spatio-temporal information, and in particular it has increased the interest of researchers on studies about user movements, behaviors and habits. Nowadays several mobile applications have been developed with the aim to exploit information extracted from raw location data. Some of those track users movement during sport activities in order to monitor their performance and to give suggestions about the next training. Other applications use GPS data to track users current position for navigation systems. Some companies use location data as a feature for social network based applications, in order to give new services to users based on their check-ins. Well known examples are Foursquare [8] that bases its entire service on users location information to give suggestions about points of interests, and Facebook [7] and Twitter [28] that allow users to add their location while posting a new message on their account, in order to add more information for other users.

The spread and popularity of this kind of mobile apps give people the possibility to track their location data in a lot of different ways, also associated to useful services, and to share with their friends this increasingly important source of information. This activity of sharing data provides in turn the additional advantage of improving the shared services offered to the community.

With these premises it is clear that there is a new important source of potentially interesting information to exploit. Whence, it is of utmost importance to design and implement an effective extraction process to get the right information from the collected raw location data. Moreover, it can be useful to envisage some post-process analysis, in order to infer additional knowledge about users. A good starting point is to recognize *important locations* for the users, i.e. *personal places of interest* (PPOIs): such places can tell a lot about their daily behavior and habits. In other words, PPOIs are places which have particular meaning for users, such as home, work, or any place where they spend a considerable amount of time during the day or which they visit frequently.

In this chapter, we focus on a novel proposal for PPOIs identification: in particular, we pay attention on how users move during their daily activities, in order to recognize the importance of places they visit according to different points of view, such as the frequency or intensity of visits. Indeed, we observed that some meaningful locations are related to users' main activities, thus they spent a lot of time in specific delimited geographic areas, such as their office or home. Other locations, instead, have been visited several time during the analyzed days, but with not the same intensity as home or office. An example of this kind of places may be the newsstand or the supermarket. In order to recognize PPOIs, we must first be able to detect the so-called *stay points* (SPs), i.e. locations where the users "may stay for a while" (see [18]). Not all stay points can be considered important places, but they are good candidates and effective off-the-shelf tools are available to extract them from raw data (whatever the source, like, e.g. a GPS-device). The candidate stay points need then to be filtered to provide the final set of PPOIs. We remark here that our proposal is technology-independent, being based only on raw data: neither we carry out any enrichments of positional data nor we use any external knowledge sources (like, e.g. georeferenced posts or

resources published on Twitter, Facebook or other social networks). As we will see in Sect. 2, Urban computing [32] and trajectory data mining [37] are two research fields which can greatly benefit from this kind of work.

In the literature, earlier approaches focus on the density of detected positions inside a delimited area, and on time thresholds to check when changing area, in order to recognize the locations which might have particular meaning for users. However, this is not enough to ensure a good selection, which should also take care to discard all “false important places” (e.g. crossing at intersections or stops at traffic lights) and, at the same time, should not miss relevant locations. Indeed, grid systems which exploit density, but are based on cells of fixed dimensions, cannot always guarantee a correct recognition due to the location distribution on the geographic space: the cell bounds might overlap an important place and, as a consequence, the latter will be divided and wrongly processed as two or more distinct places.

Further complexity comes into play since users movements are affected by other factors, such as speed/acceleration, heading, relations between locations, and also by the changes of the accuracy of GPS devices during subsequent detections. Many approaches considering the speed parameter tend to identify stay points when the measurement of speed is (nearly) zero. However, this assumption is again not enough accurate (it is sufficient to think, e.g. of a walk in a park). Therefore, to properly understand users behavior and habits it seems more appropriate to analyze their movements by considering a set of combined elements to infer the right information about the way they move.

On this basis the novelty of our approach aims at overcoming the above mentioned issues and at refining the whole identification process. First of all, our method is modular; we exploit some state-of-the-art algorithms to do an initial filtering of the raw positional data. Then, we carry out a deeper analysis, taking into account some user-related measures as further steps to refine the recognition task. Namely, we consider the area covered by a stay point, the time spent in a given location and the frequency of visits. As we will see in Sect. 5, this second phase improves the final outcome in terms of precision (paying a little cost in terms of recall). In particular, our approach allows us to infer a description of places in terms of a set of *features* more related to users routine activities. Mapping the physical locations into an abstract space based on those features helps us to carry on a deeper analysis which allows us to observe if a place is repeatedly visited. Moreover, we can identify locations (e.g. rendez-vous points, newsstands, bus stops to name a few) which are visited several times during a longer period, but not with a sufficient “intensity” to be found by previous techniques.

This chapter is structured as follows: in Sect. 2 we discuss related work. The problem statement we focus on, together with the main notions and definitions, is presented in Sect. 3, while our proposed approach is described in Sect. 4. Section 5 is devoted to the experimental evaluation and, finally, we draw conclusions and some future work directions in Sect. 6.

## 2 Related Work

### 2.1 Human Mobility

Some authors focus on analyzing patterns in mobile environments. A study, presented by Laxmi et al. [16], analyzes the behavior of user patterns related to existing works from the past few years. Noulas et al. [25] analyze a large dataset from Foursquare in order to observe user check-in dynamics and find spatio-temporal patterns. Their results are useful to study user mobility and urban spaces. In this direction other authors present their work on analysis of user communities in order to build human mobility models. Karamshuk et al. [15] survey existing approaches to mobility modeling. Hui et al. [12] propose a system to improve the understanding of the structure of human mobility by analyzing the community structure as a network. Mohbey et al. [22] propose a system based on mobile access pattern generation which has the capability to generate strong patterns between four different parameters, namely, mobile user, location, time and mobile service. They focus on mobile services exploited by users and their approach shows to be very useful in the mobile service environment for predictions and recommendations. Zheng et al. [33, 35, 36] developed a brand new social network system based on user locations and trajectories, called GeoLife, which aims to mine correlations between them.

Other researchers focus on locations analysis for destination and/or prediction of places of interest (POIs); Avasthi et al. [1] propose a system for user behavior prediction based on clustering. They analyze the differentiated mobile behaviors among users and temporal periods simultaneously in order to make use of clusters and find similarities. Zheng et al. [34] perform two types of travel recommendations by mining multiple users' GPS traces: top interesting locations and locations which match user's travel preferences. In [20] the authors combine hierarchical clustering techniques, to extract physical places from GPS trajectories, with Bayesian networks (working on temporal patterns) and custom POIs databases to infer the semantic meaning of places. Thus, they are able to discover in an effective way users PPOIs. Scellato et al. [26] developed a framework called NextPlace, a novel approach to location prediction based on time of the arrival and time that users spend in relevant places. Liu et al. [19] propose a novel POI recommendation model, exploiting the transition patterns of users' preference over location categories, in order to improve the accuracy of location recommendation. Another work in the direction of providing personalized (i.e. more accurate) POI recommendations is [3] where personalized Markov chains and region localization are used to take into account the temporal dimension and to improve the performance of the system. Finally, in [9] Gao et al. leverage on content information available in location-based social networks, relating it to user behaviour (in particular to check-in actions), to improve the performance of POI recommendation systems.

## 2.2 *Important Places Recognition*

It is clear how one of the most important issues underlying these systems is the inference of users' important places. Several studies focus on this topic to propose new approaches on important places recognition, and thus provide novel algorithms to use on more complex systems. Passing from raw information about coordinates to semantically enhanced data (landmarks or places) is an important aspect in the task of discovering important places. In [14], Kang et al. introduce a time-based clustering algorithm for extracting significant places from a trace of coordinates; moreover, they evaluate it using real data from Place Lab [27]. Hightower et al. [11] exploit WiFi and GSM radio fingerprints (collected by mobile devices) to automatically discover the places people go, associating names and/or semantics to coordinates, and detecting when people return to such places. Their BeaconPrint algorithm, according to the authors, is also effective in discovering "places visited infrequently or for short durations". De Sabbata et al. [5, 6] provide an adaptation of the well-known PageRank algorithm, in order to estimate the importance of (square) locations on the basis of their geographic features (i.e. if they are contiguous or not) and the movements of users. In particular, in the calculus of the importance (rank) of a location, the speed can be used to highlight either places where the user has stopped or places where there is a high traffic density. Thus, the notion of importance of a location can be "customized" on the basis of the current needs or situation.

Li et al. [17] mine single user movements in order to identify stay points where users spend time; then, by analyzing space and time thresholds, they compute a similarity function between users based on important places that represent them. Montoliu et al. [23, 24] propose a system based on two levels of clustering to obtain places of interest: first, a time-based clustering technique which discovers stay points, then a grid-based clustering on the stay points to obtain stay regions. Isaacman et al. [13] propose new techniques based on clustering and regression for analyzing anonymized cellular network data usage to identify generally important locations.

Many of these approaches base their algorithms on the number of user detected positions within a geographic area, and in some works with attention to the elapsed time between a detected position and the next one. For instance, in [29], Umair et al. introduce an algorithm for discovering PPOIs, exploiting a notion of "stable and dense logical neighborhood" of a GPS point. The latter is automatically determined using a threshold based approach working on space, time and density of detections. To improve the recognition process, other factors and parameters are taken into consideration to enhance the algorithms. Xiao et al. [30] add semantics to users' locations based on external knowledge (POIs databases), in order to understand user's interests and compute a similarity function between two of them without overlaps in geographic spaces. More recently Bhattacharya et al. [2] extract significant places exploiting speed and the bearing change during user movement.

An interesting approach is presented in [10] where Hang et al. present Platys, an adaptive and semisupervised solution for place recognition. Its novelty amounts to the fact that it makes minimal assumptions about common parameters (e.g. types

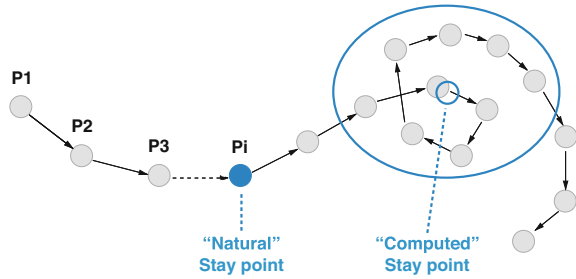
and frequencies of sensor readings, similarity metrics) which are usually tuned up manually in other systems. Instead, Platys assumes that the user visits important places sufficiently often, letting him to label the place at any time (the user is also prompted at random intervals by the system).

### 3 Problem Statement

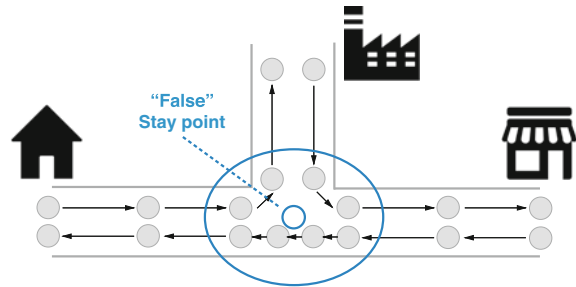
#### 3.1 Definitions

By observing a dataset of users' movements readings, it is possible to notice some coordinates where people remain stationary for long time periods, often inside buildings or delimited areas where they perform their daily activities. We call those locations *Stay Points* (SPs), as described and defined in [17]. Theoretically, a stationary user generates the same location data for all the stay time, i.e. the same point in the geographic space; we call those places *Natural Stay Points*, due to the nature of data that does not require any particular processing to understand the corresponding visited locations. However, in real situations there are several factors that affect the tracking of user movements. Due to technology limitations, there may be locations where the position detection is not possible, or the user moves in a way that the detection result cannot be so accurate. For instance, if we use a GPS, there are places where there is no signal or where the accuracy is very low due to the transportation mode that varies from underground to surface. These issues led us to have data generated by several detections which do not properly match when the user is stationary. Instead, they yielded a group of points corresponding to a location with a high density of detections within a given (limited) range. This situation may also occur when users move inside a delimited area, such as their work place where they may move among offices, or during a walk inside a mall. As described in [17], for both these latter situations we can compute the mean point of that cluster of detections in order to determine the user's stay point. We call this kind of places *Computed Stay Points*, since they approximate the original real locations. Figure 1 shows an example of user's movement readings with the two types of stay points described above. The process to identify stay points from user movements readings helps to get the set of visited locations, but neither necessarily all of them are important for the user [17] nor they provide information. By analyzing just the density of detections, some locations may be recognized as stay points even if they are not strictly related to user's main visited places. Figure 2 shows how a road crossing, where users transit a lot of time during their activities, can generate a geographic region with high density of detections, and consequently a possible stay point. We call those places *False Stay Points*, because they identify locations that do not represent a user activity, and do not provide important information about user habits and behavior.

**Fig. 1** Stay point types from user positional data



**Fig. 2** The density problem of important places discovering



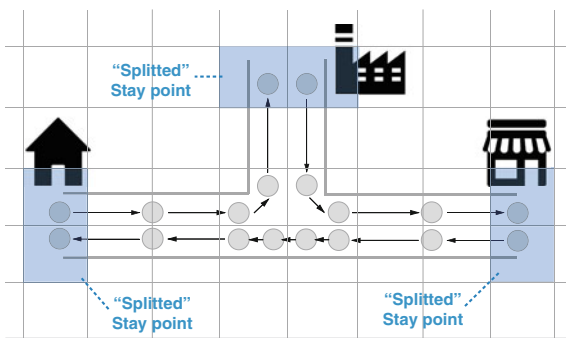
On this basis it is clear what kind of locations we consider *Important Stay Points*, namely PPOIs: locations that can help to infer information about the user who has visited them, in particular the activities that may have been carried out at each location, the stay time, and how frequently it has been visited.

### 3.2 Challenges and Motivations

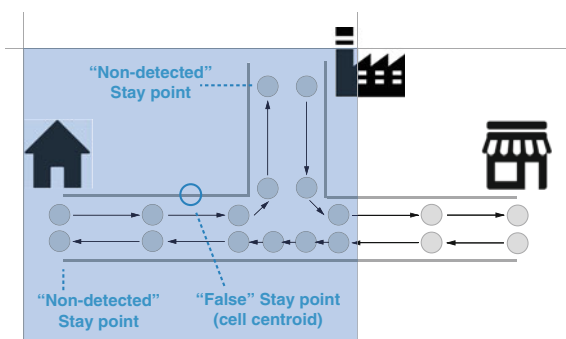
To better understand what are the main problems and difficulties emerging with important places recognition, we list a set of conceptual problems presented by the current state-of-the-art solutions. We have also run a preliminary experiment to analyze how much the conceptual problems do appear in practical scenarios, and which of them are addressed by the existing solutions; we discuss the results in detail in Sect. 3.3.

A first approach based on density may exploit a spatial subdivision of the territory where user moved to recognize the most visited locations, and consequently assign an importance value to them, but, as described in [17], this grid-based solution is affected by several issues. The cell definition during the spatial subdivision is not a technique that can be adapted to each case and to each user movement style. As represented in Figs. 3 and 4 the cell might have a size not appropriate to analyze each user and each movement, causing the not-proper recognition of PPOIs due to what we call *boundary problem*, which might divide them (Fig. 3), or include more than one of them (Fig. 4). So two first conceptual problems are:

**Fig. 3** The boundary problem of important places recognition, with too small cells (P1a)



**Fig. 4** The boundary problem of important places recognition, with too large cells (P1b)



**P1a: Boundary problem—undersized cells** With a grid based approach, cells can be too small, and thus wrongly split a stay point.

**P1b: Boundary problem—oversized cells** With a grid based approach, cells can be too large, and thus wrongly identify false stay points that either merge two or more stay points, or even are created without any real stay point.

The technique used in [17, 36] for stay point computation avoids the static approach used in the grid-based solution, which mainly analyzes the user movements as an overview on a map, in favor of a dynamic approach that scans each detected position, in order to reproduce the user movements and get more information from user behavior. By using a dataset composed of users' GPS detected positions, it is possible to avoid problems related to grid cell size by focusing on defining thresholds, based on space and time, to recognize when users move and when they remain stationary.

Let  $P = \{p_1, p_2, \dots, p_n\}$  the list of points corresponding to GPS readings ordered by time of detection,  $tT$  the time threshold and  $dT$  the distance threshold. By checking the time and distance thresholds between the point  $p_i$  and the point  $p_{i+1}$  it is possible to know if the user moved or not in that specific delimited geographic region. We call that space *segment*, since it approximates the original real user movement between the two analyzed points. If the user remains stationary (i.e. she does not exceed both the thresholds), this process can be repeated by keeping fixed  $p_i$ , scanning the



next points  $\{p_{i+2}, p_{i+3}, \dots, p_n\}$  and stopping when the thresholds are exceeded, in order to detect when and where the user changes behavior. At the end of this process it is possible to compute a *Mean Stay Point* based on the current set of analyzed points from  $p_i$  to  $p_{i+k}$ , with  $1 \leq k \leq (n - i)$ , by calculating the average latitude and longitude of points.

This technique, based on space and time thresholds, is not affected by the issues related to the cell size, which is dynamically determined, but some problems are still present (and we will indeed observe its performance in our preliminary experiment in Sect. 3.3). On straight and long trajectories, where users move with no particular changes in speed, the dynamic approach performs a scanning which, after a certain number of points, computes a stay point based on the exceeded thresholds, i.e. the mean of points in the analyzed segment, and it repeats this process for all the trajectory length, thereby determining a set of consecutive false stay points. Figure 5 shows an example of this issue displaying a path between two important places segmented with false stay points. We call this problem:

**P2: segmentation problem—constant speed** A trajectory between two distant important places is divided into several segments defined by the computed false stay points.

Other works in the literature [2, 33] introduce other parameters to use and improve the previous approach and minimize the segmentation problem. In particular they use new thresholds based on user speed, acceleration and even heading change, in order to better understand user behavior. More precisely, speed and acceleration

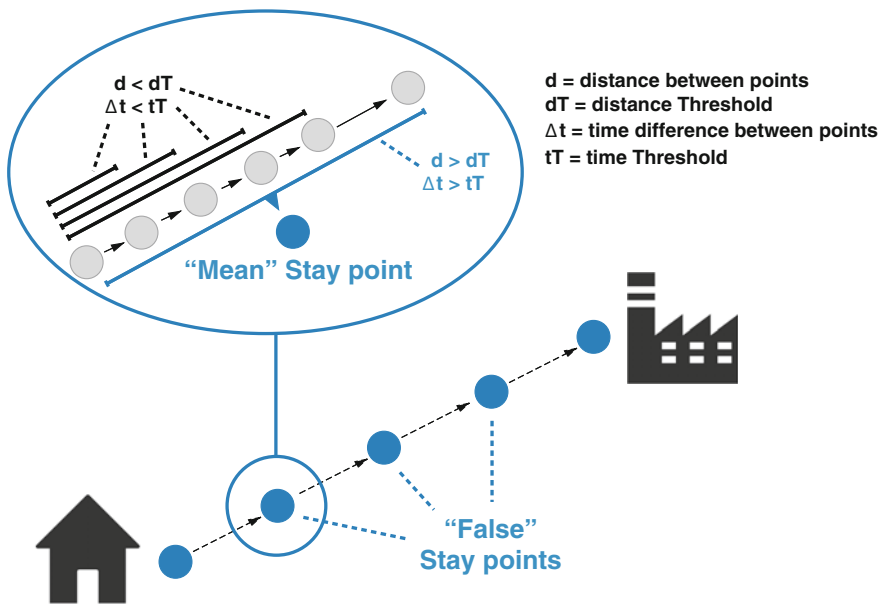


Fig. 5 The segmentation problem of important places recognition (P2)

thresholds are used in the same way as those about space and time, i.e. as soon as they are exceeded, the scanning process stops in order to compute the stay point. The heading threshold on the other hand is used in the opposite way: a constant heading indicates a movement from a SP to the next one. However, also with these approaches potentially there are some difficulties to avoid the computation of false stay points (and our preliminary experiment confirms that). For example, if a user moves with a high speed for a long time, i.e. while driving on an highway, she still exceeds the speed threshold, and after a certain amount of time the other ones, causing again the segmentation problem.

Whence, fixed thresholds may not be suitable for all user movements; indeed, some settings perfectly tuned for some users may be very wrong for others. As we will see in Sect. 5, by changing the thresholds values we observed how the recognition process varied the granularity (i.e. the number and the density of stay points) of the result, providing different set of stay points. This issue causes the computation of false stay points if the thresholds are not properly set considering the current user movements to analyze. User activities which involve several vehicles and in wider areas generate different datasets compared to users that move in small regions and mainly with one mode of transportation; whence the need of different analysis. Figures 6 and 7 show two examples where wrong thresholds raise the two last problems:

**P3a: Fixed thresholds problem—slow speed** In Fig. 6, it is possible to see how a region (delimited by the circle) where user moved with very slow speed, differently from the rest of the tracked movement, makes the threshold-based techniques unable to properly recognize the PPOIs, due to a too high threshold for the current tracked movement. Indeed, as soon as the speed exceeds the related threshold (changing from slow to high again), the whole slow speed region inside the circle will be processed in the same way of a walk inside a building, therefore generating a single false stay point. Moreover, the latter, whose position is the result of a mean of the coordinates of all the points inside the circle, can also be put in a totally wrong place, w.r.t. the progress of the path in the region.

**P3b: Fixed thresholds problem—high speed** On the other hand, Fig. 7 shows how regions with high speed (higher than the threshold set), in a trajectory between two locations, generate false stay points, again due to a not proper threshold value setup.

### 3.3 Preliminary Experiment

In order to understand the impact of problems described in Sect. 3.2 on real world user movements, and the effectiveness of the most used approaches in the literature, we have planned two evaluation tasks. The first one is based on an in-house dataset. Indeed, we built a mobile application (in two versions, for both iOS and Android smartphones) to gather real movement data from people. Basically, we needed a sequence of GPS points consisting in latitude, longitude, speed, timestamp and accuracy, to have a trajectory that represents how and where user moved. We

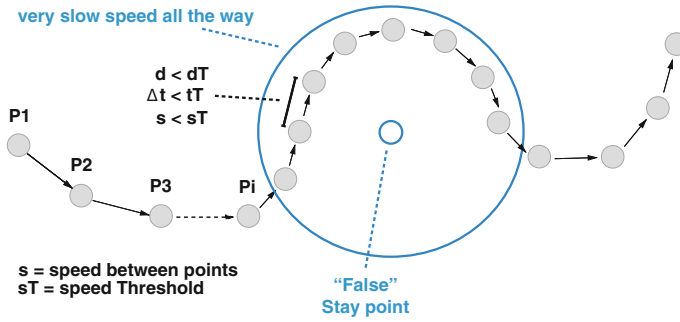


Fig. 6 The thresholds problem of important places recognition, the slow-speed issue (P3a)

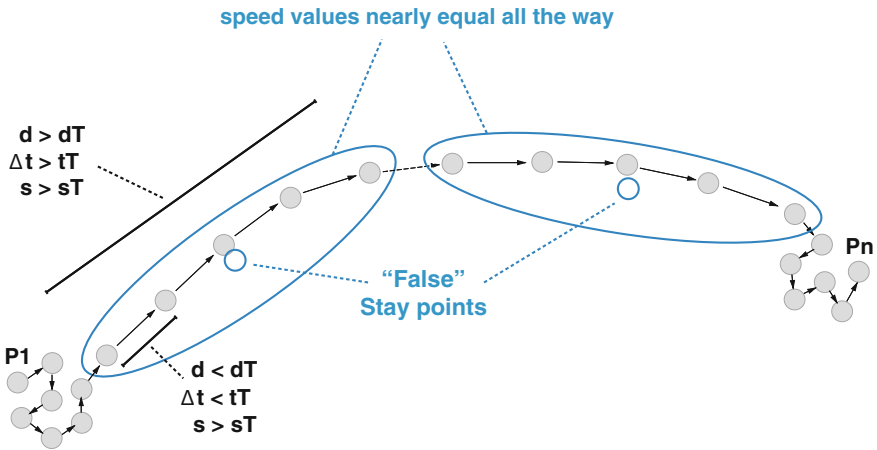


Fig. 7 The thresholds problem of important places recognition, the constant-speed issue (P3b)

have chosen a sample of 13 (Italian) users in order to collect a sufficient amount of GPS detections during 4 days of common daily activity. The second evaluation task involved the same group of 13 users, but on 4 days of movements related to 13 maps (one for each user) taken randomly from the GeoLife dataset [35]. The latter has been collected in (Microsoft Research Asia) GeoLife project by 182 users in a period of over three years (from April 2007 to August 2012: for the details see [21]).

Designing the two tasks, we paid attention to have different types of behavior, from frequent home-work travels to routines very stationary, also with different modes of transportation, e.g. motorized vehicles, bicycle, walk. We have estimated to collect (for the in-house task) and to choose (for the GeoLife task) data for 4 days for each user, in order to have enough detections to properly recognize behaviors and habits, since a lower number of days might not emphasize locations with high frequency and/or intensity.

Of course, a key difference between these two preliminary evaluation tasks is related to the users' knowledge about the datasets. In the in-house case each user evaluated the performance of the algorithms on its own data, being in the perfect condition to establish the ground truth. Instead, in the GeoLife case there was no ground truth about PPOIs available in the database, and our users were not acquainted with the Chinese regions of GeoLife. However, in each case users had the same skill and knowledge level in identifying the potential important places.

We implemented a set of popular algorithms used in the literature to check what issues affect them. The first, named  $G$ , is a static approach based on the grid method described in [17], useful to see how the boundary problem affects the results on dataset with movements from different user behaviors and habits. The second one is based on a dynamic approach and only space threshold, named  $S$ , as described in [18]. We have also implemented the  $T$  and  $V$  versions of threshold-based algorithms, since they have been often used in literature, even recently [17, 23, 29, 36]. Moreover, we have developed further versions of the latter algorithms with more parameters as thresholds, such as acceleration, and heading change, named  $A$  and  $H$ , respectively (like in [2, 33]), to see how the addition of parameters affects the PPOIs identification.

We have run all algorithms to see the results on our datasets and make some considerations about the issues explained in Sect. 3.2. Observations on results showed that:

- the static approach, namely the grid-based clustering, got variable performance due to different types of movement that need different cell sizes (P1a, P1b):
  - smaller cells allow us to recognize the right SPs, but adding a lot of false SPs;
  - larger cells generate the right number of SPs, but with wrong locations since the centroid is taken as the mean of all points in the cell;
- dynamic approaches fit well any type of movements readings;
- generally, to add new thresholds based on new parameters helped to discard false stay points;
- acceleration seems to be a too strict parameter, since too many points are discarded;
- heading change gives a low contribution to PPOIs identification, anyway it helps to improve the precision of the recognition process;
- the segmentation problem (P2) is still present;
- to use fixed thresholds does not allows us to always have a perfect setup for all situations, due to the different types of movements (P3a, P3b);
- generally, the preliminary experiment encourages us to adopt dynamic approaches exploiting several parameters with an automatic thresholds computation methodology (also helping to deal with “sensitive” parameters like, e.g. acceleration).

This preliminary experiment helped us to confirm how the above mentioned approaches still present some issues and could be improved. Figure 8 illustrates the cumulative rating distribution for all the algorithms considered in the preliminary experiment (notice that in the figure the lines for  $S$  and  $T$  algorithms coincide, the same for  $A$  and  $AH$ ). Table 1 shows the average ratings, precision, recall and F-measure reported by each algorithm. We can see that, despite the higher precision

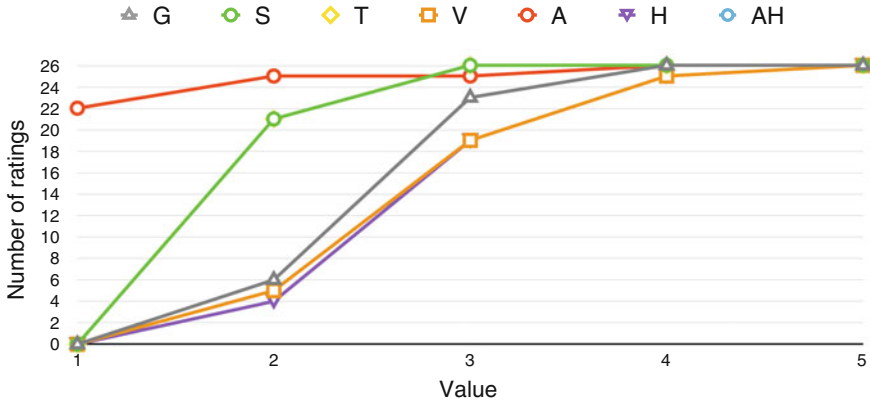


Fig. 8 Preliminary experiment: cumulative rating distribution for all algorithms

Table 1 Preliminary experiment: algorithms comparison

Algorithm	Average rating	Precision	Recall	F-measure
<i>G</i>	2.88	<b>0.186</b>	0.806	<b>0.302</b>
<i>S</i>	2.19	0.026	<b>1</b>	0.051
<i>T</i>	2.19	0.041	<b>1</b>	0.078
<i>V</i>	<b>3.11</b>	0.138	0.846	0.238
<i>A</i>	1.23	0.125	0.063	0.084
<i>H</i>	<b>3.15</b>	0.160	0.835	0.269
<i>AH</i>	1.23	0.125	0.063	0.084

and F-measure of *G* w.r.t. *V* and *H*, users have preferred the latter two algorithms with better average ratings (3.11 for *V* and 3.15 for *H* vs. 2.88 for *G*). This can be explained considering that *G* does not discard any candidate SPs, but it simply clusterizes them. Hence, the user can be confused looking at the representation in the map, seeing many “spurious” points scattered around in a uniform way. Moreover, sometimes the grid-based approach does not identify the right coordinates of important places, due to the cluster centroid which is affected by the high number of points contained in the cell (which can be too large). Algorithms *S* and *T* got the highest recall with a score equal to 1, but with very low performances in terms of precision and average rating. Finally, we ran the Wilcoxon test in order to verify if there are significant differences among the rating distributions got by the algorithms. The resulting *p*-values appear in Table 2. We can observe that there are statistical significances between several pairs of algorithms (where the *p* value < 0.005). In particular, we can confirm again that increasing the number of parameters used as thresholds by the algorithms allow us to get a significant improvement, apart the cases of the threshold *T* which does not give any contribution and the threshold *H* which contributes slightly.

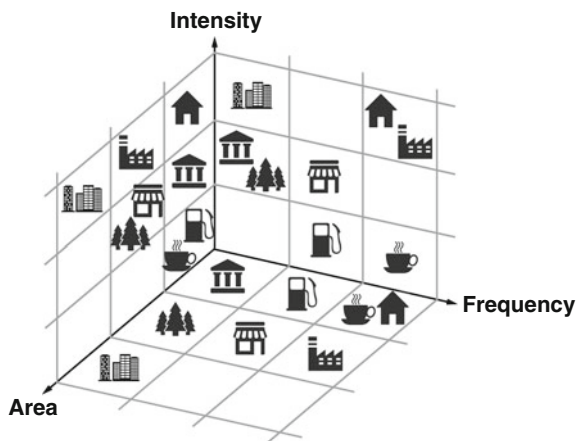
**Table 2** Preliminar Wilcoxon test:  $p$  values

	G	S	T	V	A	H
$G$	–	–	–	–	–	–
$S$	$2.467e-05$	–	–	–	–	–
$T$	$2.467e-05$	NA	–	–	–	–
$V$	0.01966	$1.507e-05$	$1.507e-05$	–	–	–
$A$	$4.732e-06$	$3.69e-06$	$3.69e-06$	$4.1e-06$	–	–
$H$	0.01073	$9.044e-06$	$9.044e-06$	1	$3.586e-06$	–
$AH$	$4.732e-06$	$3.69e-06$	$3.69e-06$	$4.1e-06$	NA	$3.586e-06$

### 4 Proposed Approach

The key contribution and novelty of our methodology for the recognition of PPOIs ultimately rely on a mapping from the physical space (determined by raw positional data) to an *abstract* space, called *features space*. The latter allows us to consider as coordinates some *features* which are semantically related to users’ habits and behaviours. Figure 9 shows an example of how important places can be positioned in our features space. The icons indicates some kind of common places, such as home, office, bar, mall and fuel station, but also a park, a travel to a city, or a visit to a museum. From a pragmatic point of view places represented in this space allow one to infer some similarities among them. In the example it is possible to see how they may be related according to a couple of features. For instance, a bar and a fuel station have low values on *area dimension* and *intensity* but they differ in *frequency*. Or, home and bar may have the same *area dimension* and the same *frequency*, but people spend a different amount of time in those locations.

**Fig. 9** Some kind of important places positioned into the features space



Thus, we can provide a deeper and more meaningful representation of PPOIs: for instance, in the following we will see how we can observe if a place is repeatedly visited or if it is visited several times during a longer period, but not with a sufficient intensity to be taken into consideration by previous techniques (this can be the case of, e.g. rendez-vous points, newsstands, bus stops etc.).

Our approach for PPOIs recognition consists in a method that analyzes a dataset of user movements readings, regardless of the type of technology used for tracking, even without any help from external knowledge sources. A point in the dataset just needs to be described by a set of coordinates to identify the location into a space, and a timestamp to understand the temporal order of detected points. On this basis it is possible to work on datasets with data gathered with various technologies, for instance WiFi triangulation inside a building, such as a mall or a museum, or by using a GPS for outdoor movements.

More precisely, we rely on a dynamic approach used in the literature (see [17, 23, 36]) which analyzes user movements point by point and identify stay points by checking thresholds, as described in Sect. 3. In this way we get a set of *Candidate Important Places*, due to the nature of user stay points which represent possible locations with particular meaning for the user. In more details, our approach is organized in two main steps, with a preliminary phase consisting in defining the values of the thresholds to use (based on the tracked user activity) as follows: the first one exploits a stay point computation algorithm to get a set of candidate important places; then the second step applies our feature-based technique to properly select the most important places for the current analyzed user. These steps are described in full details in the following sections.

We conclude this paragraph noticing that intensity, frequency or other similar features have already been taken into account in the literature. For instance, in [4] Chon et al. propose to combine external knowledge from crowdsourcing and social networks data, to automatically provide places with a meaningful name or a semantic meaning. In order to carry out this goal, they consider several factors such as *residence time* (indicating “stay behavior of users at a place tied with time-of-day”) and *stay duration* (indicating “pattern of stay behavior without time-of-day”). More in general, the very concepts of features space and feature vector have been exploited in [31], in order to find similarities between users, starting from their location histories.

#### ***4.1 Preliminary Phase: Thresholds Definition***

As preliminary phase, we address the threshold definition problem. As explained in Sect. 3, there are no fixed values for thresholds that fit perfectly for each user and for each dataset; therefore a brief reasoning may help to understand what kind of movements we are analyzing. During our preliminary experiment (see Sect. 3.3) we

observed that changing the speed threshold highly affected the results for users with different use of the vehicles and transportation mode, and even the acceleration and heading change are strictly related to how users move routinely. Therefore, we define a method for extracting a good set of values for these three thresholds. We run a scan on the dataset in order to get information about the three parameters described above, paying attention on the median of non-zero values of speed, acceleration and heading change between each couple of points. This choice stems from the considerations discussed in Sect. 3; indeed we observe that the median of all speeds reached by the analyzed user, may be a good value to identify when user changes behavior. We adopt the same consideration for acceleration and heading change, in order to have a set of thresholds to use in the next step to build algorithm variants for comparison purposes.

About distance and time we keep the thresholds fixed. We set the distance threshold  $dT$  equal to 50 m, and time threshold  $tT$  equal to 50 s. These are parameters set empirically, by observing a sample of user movements during the preliminary experiment (see Sect. 3.3), where we noticed that they do not strongly affect the stay points identification.

## 4.2 Step 1: Stay Points Computation

As second step we identify the user stay points, by using a dynamic approach which consists in a scan of all points in the dataset, in order to simulate and reproduce the movement, and exploits thresholds based on some parameters to understand user behavior and recognize when and where users move or remain stationary in a location. To make possible a proper evaluation of our method, we implement several solutions of this dynamic approach; in particular, we want to compare earlier methods based just on space and/or time to others that also exploit speed, acceleration and/or heading change. By observing the results of our preliminary experiment (see Sect. 3.3), we notice that space and time are not sufficient to properly determine the right set of stay points, and also other related works take into account other parameters [2, 33]. Moreover, acceleration and heading change were too strict as parameters of selections, in our heterogeneous dataset, and they have led the algorithm to discard too many stay points. Based on these observations, we chose to use space, time and speed parameters as thresholds for the stay point computation module. More formally, during the analysis of a point  $p_i$  and a point  $p_j$ , i.e. the next one in the user trajectory, we add the point  $p_i$  to the list of candidates for the stay point computation if one or more of the following constraints are satisfied:



$$\begin{aligned}
distance(p_i, p_j) &\leq dT \\
timeDiff(p_i, p_j) &\leq tT \\
speed(p_i, p_j) &\leq sT
\end{aligned}$$

During the computation we may also take into account the accuracy of coordinates detected during the movement tracking process. If the analyzed dataset provides the accuracy values for each point reading, it is possible to improve the parameters computation between two points. For instance, if we use a dataset with data gathered by using a GPS, we can discard coordinates with very low accuracy, in order to avoid weird values due to detection errors, or even we can exploit the instant speed detection, if the accuracy is good enough to make the value reliable. On this basis, our method checks the presence of the accuracy parameter into each entry of the dataset in order to exploit it for discarding data with low reliability, and to use the instant speed, if detected. If the dataset provides this additional information, we keep only data with *accuracy*  $\leq 30$  m,<sup>1</sup> in order to avoid errors in distance computation and user speed analysis, due to problems with point data acquisition. For the speed computation we also take into account the instant speed as follows:

$$speed(p_i, p_j) = \begin{cases} \frac{segSpeed(p_i, p_j) + iSpeed(p_i, p_j)}{2} & p_j.acc \leq 10 \\ segSpeed(p_i, p_j) & \text{otherwise,} \end{cases}$$

where  $p_j.acc$ <sup>2</sup> is the GPS accuracy value for that specific detection,  $iSpeed(p_i, p_j)$  is the average value of instantaneous speed detected by the GPS in points  $p_i$  and  $p_j$ , and  $segSpeed(p_i, p_j)$  is the average speed from the point  $p_i$  to the point  $p_j$  in the user trajectory, i.e. the space segment  $\overline{p_i p_j}$ .

If  $speed(p_i, p_j)$  is above the speed threshold, the user might be moving, thus we update the point scanning with  $i = j$ , in order to discard locations which could not be appropriate stay points. Otherwise, if user has low speed, we keep fixed  $p_i$  and perform a scan over the next points  $p_{i+k}$ , with  $1 \leq k \leq (n - 1)$ , in order to detect locations to add to the list of candidates for the stay point recognition, focusing on distance and time thresholds, but also keeping checked the speed for the scan update. When the speed threshold is exceeded again, the list of candidates is processed in order to compute a *Mean Stay Point*, and the scan can continue with the next points. Algorithm 1 illustrates the stay points computation process in detail.

---

<sup>1</sup>*Accuracy*  $\leq 30$  is a parameter set empirically, by observing the raw data.

<sup>2</sup> $p_j.acc \leq 10$  is a parameter set empirically, by observing a set of GPS detections in several signal acquisition conditions.

**Algorithm 1** SPs computation

---

**Input:** A set of user movement readings  $P = \{p_0, p_1, \dots, p_n\}$ , a distance threshold  $dT$ , a time threshold  $tT$ , and a speed threshold  $sT$

**Output:** A set of SPs  $SP$

```

1:  $i, j = 0; n = |P|; q = newPoint$ 
2:  $CP = \{p_0\}$  ▷ list of candidate points
3:  $SP = \{\}$  ▷ final list of SPs
4: while  $i < n$  do
5:    $j = i + 1$ 
6:   while  $j < n$  do ▷  $p_i, p_j \in P$ 
7:     if  $dist(p_i, p_j) > dT \ \& \ time(p_i, p_j) > tT \ \& \ speed(p_i, p_j) > sT$  then
8:        $q.coord = meanCoordInCP()$  ▷  $\forall p_k | i \leq k < j$ 
9:        $q.arrivalTime = p_i.time$ 
10:       $q.leaveTime = p_j.time$ 
11:       $SP.insert(q)$ 
12:       $i = j$ 
13:       $CP = \{p_j\}$ 
14:      break
15:     else
16:       if  $speed(p_i, p_j) \leq sT$  then
17:          $CP.insert(p_j)$ 
18:       end if
19:        $j = j + 1$ 
20:     end if
21:   end while
22: end while
23: return  $SP$ 

```

---

**Table 3** Stay points computation algorithms variants

Algo name	Thresholds
S	Space
T	Space, time
V	Space, time, speed
A	Space, time, speed, acceleration
H	Space, time, speed, heading change
AH	Space, time, speed, acceleration, heading change

With the same methods described in the algorithm for the thresholds definition and in Algorithm 1 we implemented several versions of the stay point computation algorithm based on different thresholds, in order to compare the performance and understand what are the most useful set of thresholds for user behavior analysis. Table 3 shows all variants implemented for the comparison and evaluation process.

### 4.3 Step 2: Important Places Recognition

The main idea inspiring this step of our method is to map physical locations to an *abstract* space defined by a set of features more semantically related to users' habits and behaviours. For instance, a candidate feature is the frequency of visits, since

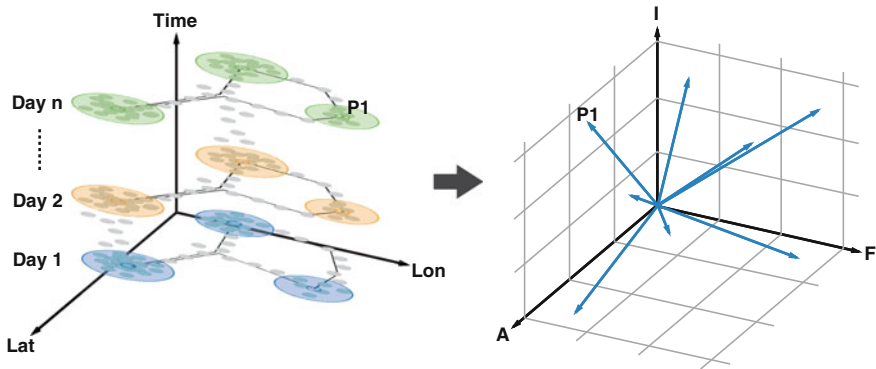


Fig. 10 The feature-based approach that moves places into the feature space

users tend to behave similarly in everyday’s life. Thus, in order to define a procedure for the important places recognition, it is useful to observe users’ movements across a period of time longer than a single day.<sup>3</sup> In other words we want to explore the possibility of superimposing the locations visited by user several times in order to extract additional semantic information, and possibly refining the results of the previous phase. Hence, to implement such strategy, we consider new parameters to describe locations, alongside latitude, longitude and timestamp, that may help to improve the recognition process.

First, we modeled the user movements readings into a three-dimensional space where a point is described by the three original raw data gathered by sensors (latitude, longitude and timestamp), in order to have a distribution of points into this space that reproduces the original user movements (see Fig. 10 (left)). We observed how the data is divided into groups, nearly in layers, which approximately represent the days when user performed the activity. Therefore this aspect makes possible further analysis and helps to get more information form each locations. On this basis we define a set of three features to describe each important place (PPOI) as a vector  $PPOI = \langle A, I, F \rangle$ , where  $A$ , the *Area* of the PPOI, is a value which indicates the diagonal extension of the rectangular region that spans over all points involved in the stay point computation. As explained in Sect. 3, when users visit locations tend to not stay perfectly stationary, but to move around a delimited area. We also keep the set of physical coordinates which describe it, in order to also represent it graphically for user-testing purposes, and for checking potential overlaps. The feature  $I$ , the PPOI *Intensity*, is a value which indicates how many times the user position has been detected inside the PPOI’s area. Finally, the feature  $F$ , the PPOI *Frequency*,

<sup>3</sup>Otherwise, activities of a single day may escape from the usual routine and could easily hinder the recognition process.

indicates how many times that location has been visited by the user, thus a parameter that increments its value each time the user came back for another visit in that place.

Formally, we have the following map ( $PhysSpace_{SP}$  is the physical space and  $FeatSpace_{SP}$  is the features space):

$$\begin{aligned}
 &feature : PhysSpace_{SP} \longrightarrow FeatSpace_{SP} \\
 &feature : \langle sp.latitude, \quad \quad \quad \langle sp.Area, \\
 &\quad \quad \quad sp.longitude, \quad \longmapsto \quad sp.Intensity, \\
 &\quad \quad \quad sp.timestamp \rangle \quad \quad \quad sp.Frequency \rangle
 \end{aligned}$$

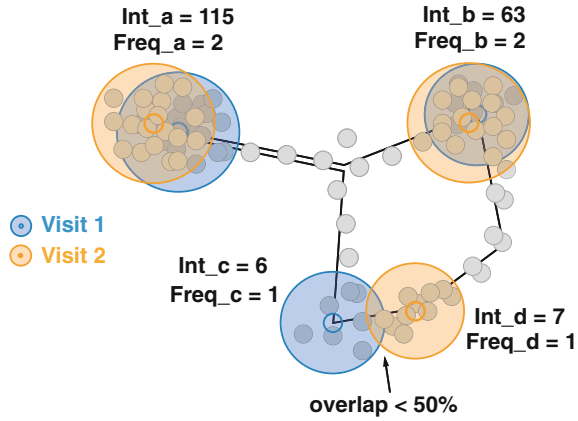
Figure 10 shows how we model movements data into the three-dimensional space, defined by *latitude*, *longitude* and *timestamp*, and how we map stay points from the physical space to the features space based on the new three dimensions *A*, *I* and *F*.

This feature-based approach makes possible a refinement step to emphasize locations visited intensively and/or repeatedly, and also to filter out the false stay points that the previous phase was not able to discard. In this phase we first analyze the SPs computed during the previous step, in order to calculate the stay point area *A* based on the detections involved in the stay point computation. Then we extract the number of detections inside that area to compute the value of *I*, and get the PPOI intensity, and we also set the frequency of each SP to 1 as initial value. With all SPs described into our features space we can analyze both the *A* and *I* distribution over the entire dataset, in order to better understand user behavior and define new thresholds that may help to filter out some places not important for that user. We set an area threshold  $aT$  equal to 3 km to run a pre-filtering process which discards SPs with diagonal area  $\geq 3$  km.<sup>4</sup> This operation helps to identify the kind of false stay points described in Sect. 3 as problems P3a and P3b, where a wrong threshold set may cause SPs generated by detections that span over a wide space. We also observed how a different use of vehicles and mode of transportation generate different density of detections, with the consequence of having higher *I* values for users that usually move slower. This issue led us to define an intensity threshold  $iT$  in order to discard SPs with *I* too low in proportion to the values obtained in the rest of the dataset. Moreover, with particular attention on the *I* values of adjacent SPs to recognize where the segmentation problem may have occurred (see Sect. 3). On this basis, we define the intensity threshold  $iT = \max3consec(intensities)$ , where *intensities* is the array with all intensity values of each SP, and the method *max3consec* returns the maximum value of intensity that in the SPs sequence is present at least three times in a row (up to some tolerance threshold for dealing with measurement errors and small deviations<sup>5</sup> from the maximum value). This technique helps us to recognize

<sup>4</sup> $aT \geq 3$  is a parameter set empirically, by observing user movements during our preliminary experiment described in Sect. 3.3.

<sup>5</sup>For instance, the user slightly changes speed while driving along a highway.

**Fig. 11** Example of overlapping activities



where the user is moving, and also where he is generating the same intensity values. By selecting the maximum value, we can discard the false stay points induced by the scenario described in the conceptual problem P2. Moreover, automatically computing the intensity threshold as previously described, we avoid locations where users stopped just once and for an amount of time not so remarkable as the time spent in home, office, supermarket, etc. Such places may be intersections with traffic lights which block vehicles for a long time, traffic-clogged streets, or rail crossings. Based on these two thresholds we run a pre-filtering process, to have a more accurate subset of SPs and proceed to take into account the frequency of visits.

By analyzing SPs sequentially, by timestamp, it is possible to check if their rectangular areas overlap, in order to get information about locations visited repeatedly. If the areas of two locations overlap with an intersection region  $\geq 50\%$ <sup>6</sup> of one of the current analyzed areas, they may be considered to represent the same place. In Fig. 11 it is possible to see an example of two visits on the same geographic area where for the locations *a* and *b* there are very similar detections on both days, therefore they represent the same important place. In that case the intensity values will be summed, the area will be their union, and the frequency will get a value equal to 2 because of the number of visits. Otherwise, the locations *c* and *d* have detections with an area overlap < 50%, therefore they will be considered as two separated places. After this filtering step, we repeat the process of merging areas several times until we get just separated regions, which identify our important places. As final phase, we run again the filtering process in order to clean out PPOIs that may have been generated with too large areas. All phases of our method named AIF are illustrated in Algorithm 2.

<sup>6</sup> *Overlap*  $\geq 50\%$  is a parameter set empirically, by observing user movements during the preliminary experiment.

---

**Algorithm 2** AIF computation
 

---

**Input:** A set of user stay points  $SP = \{sp_1, sp_2, \dots, sp_n\}$   
**Output:** A set of important places  $PPOI$

```

1:  $aT, iT, fT = 0; q = newPoint$ 
2:  $areas, intensities = \{\}$ 
3:  $PPOI = \{\}$ 
4: for  $sp_i$  in  $SP$  do
5:    $insertInAreas(sp_i.computeArea())$ 
6:    $insertInIntensities(sp_i.computeIntensity())$ 
7:    $sp_i.freq = 1$ 
8: end for
9:  $aT = 3$ 
10:  $iT = max3consec(intensities)$ 
11:  $PPOI = preFiltering(SP, aT, iT, areas, intensities)$ 
12:  $overlaps = true$ 
13: while  $overlaps == true$  do
14:    $overlaps = false$ 
15:   for  $p_i$  in  $PPOI$  do
16:     for  $p_j$  in  $PPOI \setminus \{p_i\}$  do
17:       if  $overlap(p_i, p_j)$  then
18:          $overlaps = true$ 
19:          $q.area = mergePointsAreas(p_i.area, p_j.area)$ 
20:          $q.intensity = p_i.intensity + p_j.intensity$ 
21:          $q.frequency = p_i.frequency + p_j.frequency$ 
22:          $PPOI.add(q)$ 
23:          $PPOI.remove(p_i, p_j)$ 
24:         break
25:       end if
26:     end for
27:   if  $overlaps == true$  then
28:     break
29:   end if
30: end for
31: end while
32: for  $p_i$  in  $PPOI$  do
33:    $PPOI = postFiltering(PPOI, aT, iT, areas, intensities)$ 
34: end for
35: return  $PPOI$ 

```

▷ arrays with all values  
 ▷ final list of important places  
 ▷ pre-filtering  
 ▷ empirically set to remove SPs with area diagonal > 3km  
 ▷ the maximum value repeated at least three times in a row  
 ▷ to check overlaps during the points scan  
 ▷ points merging  
 ▷ A  
 ▷ I  
 ▷ F  
 ▷ post-filtering

---

## 5 Experimental Evaluation

### 5.1 Experimental Design

To evaluate several approaches to important places identification, and to benchmark our proposed solution, we run a set of algorithms over the GeoLife dataset. We implemented the set of threshold-based algorithms described in Sect. 4, to make possible a comparison among the approaches used in the literature. As final algorithm, we have implemented our solution *AIF*, as defined in Sect. 4, to compare it to the other approaches. We have selected a sample of 16 people to evaluate the results, distributed as follows:

- 62 % men, 38 % women (all Italians);
- 62 % with age between 21 and 30, 38 % more than 30;
- 88 % with very good familiarity with smartphones;

- 56 % with intensive use of map services, 25 % with intermediate use and 19 % occasional user.

As in the preliminary experiment, all of them were not familiar with the geographic regions in GeoLife, due to the different nationality: GeoLife data have been collected in China, while our participants were Italians. Since the GeoLife dataset does not contain ground truth about PPOIs, this fact yielded the positive effect that all the participants had the same skill and knowledge level in identifying the potential important places.

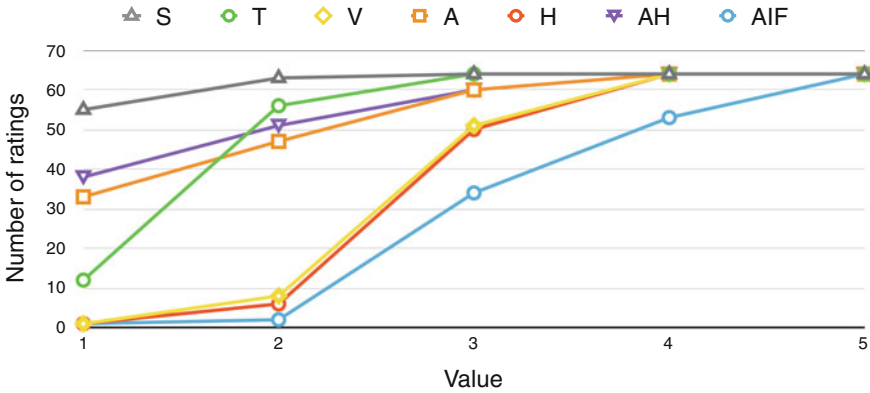
We have defined a test protocol providing detailed instructions to participants so as to guide them during the evaluation in the definition of the aspects to take into consideration. We have implemented a testing tool for them to show on a map some randomly selected sets of GPS detections from GeoLife dataset, with attention on choosing at least four consecutive days of movements readings. The participants had available a heatmap to better understand the original user movement and properly evaluate the PPOIs showed as pins on the map. The tool displays sequentially and randomly maps with pins computed by one of the algorithms previously described, in order to make not clear to participants how to associate the algorithms with the corresponding suggestions. This is a precaution to not affect them with clues during the test. During an evaluation a number between 1 and 5 indicates how they judge the overall PPOIs identification. The meanings of the rate values are the following:

1. SPs retrieved  $\leq 20\%$ ;
2. SPs retrieved  $> 20\%$  and  $\leq 50\%$  or a very high number of false SPs;
3. SPs retrieved  $> 50\%$  and  $\leq 80\%$  or  $> 80\%$ , but with an high number of false SPs;
4. SPs retrieved  $> 80\%$  and  $\leq 90\%$  and zero or a very low number of false SPs;
5. SPs retrieved  $> 90\%$  and zero or a very low number of false SPs.

Moreover, they were requested to indicate which pins properly represent PPOIs visited during the tracked activity, and also how many have been missed: this allows to compute Precision, Recall, and F-measure (i.e. the harmonic mean of Precision and Recall) of each algorithm.

## 5.2 Results

Results are reported in Fig. 12 and Table 4. The figure shows the cumulative distribution of the ratings obtained by each approach; the table shows, besides the average rating for each algorithm, also its precision, recall, and F-measure. The rating distribution and the average ratings show how the *S* algorithm obtained many 1-value ratings, due to the low filtering that it applies with the single threshold approach, thus getting a mean rate equal to 1.16; the *T* solution has been evaluated slightly better but most of rates still remains low; the adding of speed improved the performance as we expected; the *A* algorithm instead has worsened the identification process due to the acceleration parameter which has made too strict the PPOIs recognition process;



**Fig. 12** Cumulative rating distribution for all algorithms for important places identification

**Table 4** Algorithms comparison

Algorithm	Average rating	Precision	Recall	F-measure
<i>S</i>	1.16	0.004	<b>0.992</b>	0.007
<i>T</i>	1.94	0.009	<b>0.992</b>	0.019
<i>V</i>	3.06	0.126	0.657	0.211
<i>A</i>	1.81	0.286	0.217	0.247
<i>H</i>	3.11	0.131	0.657	0.219
<i>AH</i>	1.67	0.172	0.161	0.166
<i>AIF</i>	<b>3.59</b>	<b>0.370</b>	0.606	<b>0.459</b>

*H*, based on the heading change parameter, obtained a good performance but also a minimal improvement over *V*; the algorithm *AH* has been penalized by the use of acceleration; finally, our proposed method *AIF* collected a lot of positive evaluations, obtaining a mean rate equal to 3.59, the highest score among all the compared algorithms.

The simpler methods, such as *S* and *T*, got the higher recall values but with Precision very low, due to the filtering process that discards few false stay points, and provides a final set of PPOIs not so much different from the original set of movements readings. By adding more parameters as thresholds, the identification process improved, providing more accurate set of PPOIs. Moreover, the introduction of an automatic threshold algorithm computation has further improved the results. *V* and *H* solution increased the Precision, also keeping good Recall values. But the use of acceleration has reduced a lot the Precision of algorithms *A* and *AH*, obtaining very low performance in every aspect. The *AIF* solution has proven to be the most accurate method, with the highest precision and a good Recall, obtaining a good overall evaluation with the highest F-measure. The results confirm how *AIF* has improved the PPOIs identification process by providing few false stay points,



**Table 5** Wilcoxon test:  $p$  values

	S	T	V	A	H	AH
<i>S</i>	–	–	–	–	–	–
<i>T</i>	1.6e–12	–	–	–	–	–
<i>V</i>	1.415e–13	6.343e–14	–	–	–	–
<i>A</i>	4.623e–07	0.1731	2.778e–11	–	–	–
<i>H</i>	1.109e–13	1.364e–13	0.1489	1.903e–11	–	–
<i>AH</i>	2.928e–06	0.004182	5.605e–12	0.003353	5.251e–12	–
<i>AIF</i>	1.317e–12	1.086e–12	5.844e–09	1.364e–13	2.752e–08	1.107e–13

and guaranteeing not to lose too many important locations. Moreover, it provides a less confusing visualization on the map, and it is less affected by the type of users' movements.

### 5.3 Statistical Significance

Moreover, we have run a statistical test to determine whether there are any significant differences between the means of ratings got by the algorithms. Due to the nature of the data with non-normal distribution we run the Wilcoxon test in order to verify if datasets have significant differences.

Table 5 shows all the resulting  $p$ -values for each couple of algorithms to compare. We can observe that the most of them got very low  $p$ -value, lower than the standard Wilcoxon threshold 0.05. Therefore, this output indicates a statistically significant difference between means, and consequently a relevant improvement in performance for those algorithms. It is possible to notice how the low performances of *T* and *A* are not statistically different. Finally, the use of the heading change threshold did not bring a significant improvement when used in algorithm *V*, and it provided only a small noticeable improvement when used in algorithm *A*.

## 6 Conclusions and Future Work

In this chapter we have presented our proposal of important locations identification. We have stated the most common issues related to the recognition process, then we have described our approach, that consists in a new model based on a space transition from physical space to a features space, where locations are described by a set of features more related to users' habits and behaviors. The experiment performed in this work has demonstrated that the proposed approach results more effective than other related works in terms of performance, and also in difficult situations, where

other algorithms are affected by the problems described in the chapter. Moreover, the feature-based approach allows us to add more semantic value to important places, providing new information that future works may exploit for locations classification and similarity computation. For future work, we plan to work on the features space in order to explore the possibility to expand the features set and design a locations classifier based on this approach. Moreover, we want to analyze user movement types, in particular what kind of vehicles people use, and what pace they have while walking or running, in order to provide new data and further improve the important locations identification. Finally, it would be interesting to take into account the analysis of places co-located inside a single building or within a small area.

## References

1. Avasthi, S., Dwivedi, A.: Prediction of mobile user behavior using clustering. In: Proceedings of TSPC13, vol. 12, p. 14th (2013)
2. Bhattacharya, T., Kulik, L., Bailey, J.: Extracting significant places from mobile user GPS trajectories: a bearing change based approach. In: Proceedings of ACM SIGSPATIAL GIS 2012, pp. 398–401. ACM (2012)
3. Cheng, C., Yang, H., Lyu, M.R., King, I.: Where you like to go next: successive point-of-interest recommendation. In: Proceedings of IJCAI'13, pp. 2605–2611. AAAI Press (2013)
4. Chon, Y., Kim, Y., Cha, H.: Autonomous place naming system using opportunistic crowdsensing and knowledge from crowdsourcing. In: Proceedings of ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN), pp. 19–30. IEEE (2013)
5. De Sabbata, S., Mizzaro, S., Vassena, L.: Spacerank: using pagerank to estimate location importance. In: Proceedings of MSOda08, pp. 1–5 (2008)
6. De Sabbata, S., Mizzaro, S., Vassena, L.: Where do you roll today? Trajectory prediction by spacerank and physics models. Location Based Services and TeleCartography II. LNCS, pp. 63–78. Springer, Berlin (2009)
7. Facebook check-in: Who, what, when, and now...where. <https://www.facebook.com/notes/facebook/who-what-when-and-nowwhere/418175202130> (2014)
8. Foursquare check-in: About. <https://foursquare.com/about> (2014)
9. Gao, H., Tang, J., Hu, X., Liu, H.: Content-aware point of interest recommendation on location-based social networks. In: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI'15), pp. 1721–1727. AAAI Press (2015)
10. Hang, C.W., Murukannaiah, P.K., Singh, M.P.: Platys: user-centric place recognition. In: AAAI Workshop on Activity Context-Aware Systems (2013)
11. Hightower, J., Consolvo, S., LaMarca, A., Smith, I., Hughes, J.: Learning and recognizing the places we go. In: Proceedings of UbiComp 2005: Ubiquitous Computing, pp. 159–176. Springer (2005)
12. Hui, P., Crowcroft, J.: Human mobility models and opportunistic communications system design. Philos. Trans. R. Soc. A Math. Phys. Eng. Sci. **366**(1872), 2005–2016 (2008)
13. Isaacman, S., Becker, R., Cáceres, R., Kobourov, S., Martonosi, M., Rowland, J., Varshavsky, A.: Identifying important places in peoples lives from cellular network data. In: Pervasive Computing, pp. 133–151. Springer (2011)
14. Kang, J.H., Welbourne, W., Stewart, B., Borriello, G.: Extracting places from traces of locations. In: Proceedings of the 2nd ACM International Workshop on Wireless Mobile Applications and Services on WLAN Hotspots, pp. 110–118. ACM (2004)
15. Karamshuk, D., Boldrini, C., Conti, M., Passarella, A.: Human mobility models for opportunistic networks. IEEE Commun. Mag. **49**(12), 157–165 (2011)

16. Laxmi, T.D., Akila, R.B., Ravichandran, K., Santhi, B.: Study of user behavior pattern in mobile environment. *Res. J. Appl. Sci. Eng. Technol.* **4**(23), 5021–5026 (2012)
17. Li, Q., Zheng, Y., Xie, X., Chen, Y., Liu, W., Ma, W.Y.: Mining user similarity based on location history. In: *Proceedings of ACM SIGSPATIAL GIS 2008*, p. 34. ACM (2008)
18. Lin, M., Hsu, W.J.: Mining gps data for mobility patterns: a survey. *Pervas. Mobile Comput.* **12**, 1–16 (2014)
19. Liu, X., Liu, Y., Aberer, K., Miao, C.: Personalized point-of-interest recommendation by mining users' preference transition. In: *Proceedings of ACM CIKM 2013*, pp. 733–738. ACM (2013)
20. Lv, M., Chen, L., Chen, G.: Discovering personally semantic places from gps trajectories. In: *Proceedings of ACM CIKM 2012*, New York, USA, pp. 1552–1556. ACM (2012)
21. Microsoft Research Asia: GeoLife project. <http://research.microsoft.com/en-us/downloads/b16d359d-d164-469e-9fd4-daa38f2b2e13/> (2012)
22. Mohbey, K.K., Thakur, G.: User movement behavior analysis in mobile service environment. *Br. J. Math. Comput. Sci.* **3**(4), 822–834 (2013)
23. Montoliu, R., Blom, J., Gatica-Perez, D.: Discovering places of interest in everyday life from smartphone data. *Multimed. Tools Appl.* **62**(1), 179–207 (2013)
24. Montoliu, R., Gatica-Perez, D.: Discovering human places of interest from multimodal mobile phone data. In: *Proceedings MUM 2010*, p. 12. ACM
25. Noulas, A., Scellato, S., Mascolo, C., Pontil, M.: An empirical study of geographic user activity patterns in foursquare. *ICWSM 11*, 70–573 (2011)
26. Scellato, S., Musolesi, M., Mascolo, C., Latora, V., Campbell, A.T.: Nextplace: a spatio-temporal prediction framework for pervasive systems. In: *Pervasive Computing*, pp. 152–169. Springer (2011)
27. Schilit, B., LaMarca, A., Borriello, G., Griswold, W., McDonald, D., Lazowska, E., Balachandran, A., Hong J., Iverson, V.: Challenge: ubiquitous location-aware computing and the place lab initiative. In: *Proceedings of the 1st ACM International Workshop on Wireless Mobile Applications and Services on WLAN (WMASH 2003)*, San Diego, CA, September 2003
28. Twitter check-in: How to tweet with your location. <https://support.twitter.com/entries/122236-how-to-tweet-with-your-location> (2014)
29. Umair, M., Kim, W.S., Choi, B.C., Jung, S.Y.: Discovering personal places from location traces. In: *Proceedings of ICACT'14*, pp. 709–713. IEEE (2014)
30. Xiao, X., Zheng, Y., Luo, Q., Xie, X.: Finding similar users using category-based location history. In: *Proceedings of ACM SIGSPATIAL GIS 2010*, pp. 442–445. ACM (2010)
31. Xiao, X., Zheng, Y., Luo, Q., Xie, X.: Inferring social ties between users with human location history. *J. Ambient Intell. Humaniz. Comput.* **5**(1), 3–19 (2014)
32. Zheng, Y., Capra, L., Wolfson, O., Yang, H.: Urban computing: concepts, methodologies, and applications. *ACM Trans. Intell. Syst. Technol.* **5**(3), 38 (2014)
33. Zheng, Y., Li, Q., Chen, Y., Xie, X., Ma, W.Y.: Understanding mobility based on gps data. In: *Proceedings of UbiComp'08*, pp. 312–321. ACM (2008)
34. Zheng, Y., Xie, X.: Learning travel recommendations from user-generated gps traces. *ACM TIST* **2**(1), 2 (2011)
35. Zheng, Y., Xie, X., Ma, W.Y.: GeoLife: a collaborative social networking service among user, location and trajectory. *IEEE Data Eng. Bull.* **33**(2), 32–39 (2010)
36. Zheng, Y., Zhang, L., Xie, X., Ma, W.Y.: Mining interesting locations and travel sequences from GPS trajectories. In: *Proceedings of the WWW'09*, pp. 791–800. ACM (2009)
37. Zheng, Y., Zhou, X.: *Computing with Spatial Trajectories*. Springer, Berlin (2011)