

Discovering Human Places of Interest from Multimodal Mobile Phone Data *

Raul Montoliu
Institute of New Image Technologies (iNIT)
Jaume I University. Campus Riu Sec s/n 12071
Castellón, Spain
Dept. Ingeniería y Ciencia de los Computadores
Jaume I University. Campus Riu Sec s/n 12071
Castellón, Spain
montoliu@uji.es

Daniel Gatica-Perez
Idiap Research Institute.
CH-1920 Martigny, Switzerland
École Polytechnique Fédérale de Lausanne
(EPFL)
CH-1015 Lausanne, Switzerland
gatica@idiap.ch

ABSTRACT

In this paper, a new framework to discover places-of-interest from multimodal mobile phone data is presented. Mobile phones have been used as sensors to obtain location information from users' real lives. Two levels of clustering are used to obtain places of interest. First, user location points are grouped using a time-based clustering technique which discovers stay points while dealing with missing location data. The second level performs clustering on the stay points to obtain stay regions. A grid-based clustering algorithm has been used for this purpose.

To obtain more user location points, a client-server system has been installed on the mobile phones, which is able to obtain location information by integrating GPS, Wifi, GSM and accelerometer sensors, among others. An extensive set of experiments have been performed to show the benefits of using the proposed framework, using data from the real life of 8 users over 5 continuous months of natural phone usage.

Categories and Subject Descriptors

H.5.1 [Multimedia Information Systems]: Evaluation/methodology;
J.4 [Social and Behavioral Sciences]: Sociology

General Terms

Algorithms, Human factors

Keywords

Mobile phones, Multimodal data, Place discovering, real-life long-term experiments

*This work was supported by Nokia Research Center Lausanne (NRC) through the LS-CONTEXT project. R. Montoliu was also supported by the Spanish *Ministerio de Ciencia e Innovación* under project Consolider Ingenio 2010 CSD2007-00018 and grant JC2009-00038.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

MUM '10, December 1-3, 2010, Limassol, Cyprus.
Copyright ©2010 ACM 978-1-4503-0424-5/10/12 \$10.00

1. INTRODUCTION

Mobile phones are emerging as an attractive option for large-scale sensing of human behaviors. Modern smartphones are equipped with a variety of sensors, including GPS, accelerometers, Wifi and Bluetooth among others. Furthermore, almost everybody carries a mobile phone everyday. For instance, in many European countries, there are more mobile phone accounts than inhabitants and this situation is similar in many other countries. Therefore, the use of data provided from mobile phone sensors, instead of using custom-made sensor architectures, is an interesting option to study human behaviors.

The automatic learning of places of interest is one of the key tasks when studying human behavior and for designing new applications for wearable computers including mobile phones. Several potential applications could be developed for associating a to-do list with each learned location [11], for predicting user movements [1], for helping mobile devices to make decisions on how to behave based on their location [6] (e.g. the phone could automatically switched to a silent mode when the user enters a place where the ring is inappropriate), for inferring people's transportation modes [12, 17, 18, 13], for mining individual life patterns [15], and for recommending locations and activities [16, 10], among others.

In order to clarify some terms used in this paper, the concepts of *Location Point*, *Stay Point*, and *Stay Region* are defined as follows:

- A *Location Point* is a measurement provided by the sensor about the location of a user (e.g. the GPS coordinates) together with the time when the sensor captures the location information. A location point is represented using the location coordinates together with the time where the sensor captures the location information, e.g. $([46.6N, 6.5E], [16 : 34 : 57])$.
- A *Stay Point* is a cluster of location points (from the same day) which represents a geographic region in which the user stayed for a while. A stay point is represented using the coordinates of the centroid of the cluster and the time moments where the user arrived and left the stay point, e.g. $([46.6N, 6.5E], [16 : 30 : 00], [17 : 54 : 34])$.
- A *Stay Region* is a cluster of stay points (from several days) with the same semantic meaning. A stay region is represented using the coordinates of the centroid of the cluster and the minimum and maximum coordinates of the stay points

belonging to the cluster, e.g. $([46.6N, 6.5E], [46.595N, 46.599N], [6.498E, 6.502E])$. Therefore, a stay region can be represented by using a rectangle centered at the centroid of the cluster whose size depends on the minimum and maximum coordinates.

In this paper, stay region and place of interest are synonymous.

As it has been previously exposed in some previous works ([1], [6]), for most location systems, including GPS, multiple measurements in the same location do not necessarily yield the exact same coordinates due to errors and variations in the measured phenomena. This is true even if the user stops for a while at precisely the same point every day. Therefore, two estimated stay points could have the same semantic meaning (e.g. “work”, “home”, “museum”, etc.) but not necessarily the same exact coordinates. To deal with this problem, the concept of stay region is used.

Figure 1 shows an illustrative example of the three concepts. In the left map, the location points extracted for a hypothetical user visiting an European city are shown. Each color corresponds to a different day. Note that the paths followed by the user are different in the two days. The user stayed in the two days in the areas of the *Parc Olympique* and the *Lausanne train station*. One of the days (blue), the user also stayed in the area of the *Parc de Milan*. The middle map shows the stay points estimated for each day using as input data the location points shown in the left map. Note that the location of the stay points do not fall in the same coordinates, but they represent the same semantic meaning, i.e. *Parc Olympique* and *Lausanne train station*. The right map shows the stay regions estimated using the previous stay points.

In this paper, an algorithm to accurately estimate stay points and stay regions from the real life of users from mobile phone sensor data is presented. The maximum size of the region and the minimum time that a user must be in this region are controlled using two parameters. In the proposed algorithm, an important constraint has been introduced, which consists in that between two consecutive location points their time difference must also be bounded. The use of this new constraint arises from the use in this work of data obtained from users’ real lives. Thanks to this constraint all the consecutive location points included in a stay point are close in time. To estimate stay regions from stay points, a clustering algorithm must be used. In this paper, two clustering algorithms have been studied. The first one is a density-based method [3], previously used in [15], and the second one is a grid-based one presented in [16].

One of the most important aspects of this work is that mobile phones are used to obtain user location data, instead of custom-made architectures or professional GPS sensors. The use of mobile phones for large-scale sensing of human activity is an interesting alternative since people wear mobile phones almost all the time and therefore data can be obtained in a natural manner and without the necessity to wear another device, which could be uncomfortable to the user or unrealistic [2]. In addition, the data obtained by mobile phones reflects better the real life of users, since the phones used in the experiments reported in this paper were their real (and unique) phones. However, one important drawback of using mobile phones is that location data might not be obtained everytime, since the GPS sensors included in mobile phones do not provide the same quality as professional dedicated GPS sensors. Then, when using mobile phones there are more missing location data due to sensor failure

to obtain GPS coordinates.

To collect data, a novel client-server system has been used, which records GSM, GPS, Bluetooth, Wifi and motion, in conditions that are feasible for large-scale sensing and comfortable for users. The client component of the system is installed in the user mobile phone to collect data. To save battery and also to provide more location information where GPS might not be available, the system scans Wifi access points (APs) to build a private library of geocoded base stations, allowing future location observations when the user’s phone observes a previously learned Wifi AP. In addition, it is also possible to recognize periods of time when the phone is static using the accelerometer and other sensors. Thanks to the use of this system and also to the way in which the information provided by the system has been used in this work, more location points can be obtained with respect to the case of only using the GPS sensor, thus allowing for large-scale sensing along normal user days.

This paper has three main contributions:

1. We propose a new framework to accurately estimate places of interest from users real life location data. The proposed framework deals with missing location data, an inevitable problem when working with data from users’ real lives.
2. We manage the information obtained by the use of a robust sensing system which provides location data in conditions that are feasible for large-scale sensing and comfortable for users. Our approach is multimodal since location information is obtained from multiple sensors such as GPS, Wifi APs, GSM, accelerometer.
3. We conduct extensive experiments to evaluate the effectiveness of the proposed approach, using real-life mobile phone location data collected by 8 volunteers over a long-term period of 5 consecutive months.

The rest of the paper is organized as follows: Section 2 provides a discussion of the previous works in the area of learning and recognition of places of interest. In Section 3, an overview of the proposed approach is presented. Section 4 describes how location data can be obtained by using mobile phones and discusses the proposed system for sensing data. In Section 5, the algorithms to estimate places of interest are explained in detail. The experiments performed to validate our work are discussed in Section 6. Finally, the conclusions are drawn in Section 7.

2. RELATED WORK

In this section, we review two families of methods to estimate places of interest, geometry-based and fingerprint-based.

2.1 Geometry-based methods

Geometry-based techniques use location data (as GPS) to produce coordinates, circles, or polygons to describe the significant places where the user goes. These algorithms take a history of location points and find places where the person stays for significant periods of time using a clustering-based technique. The main drawbacks of GPS include its inability to function well indoors, its occasional lack of accuracy due to the geometry of visible satellites, and the loss of signal in urban canyons and other “shadowed” areas. Early work on place learning from GPS used the loss of signal to infer the location of important indoor places as buildings [11, 1]. These

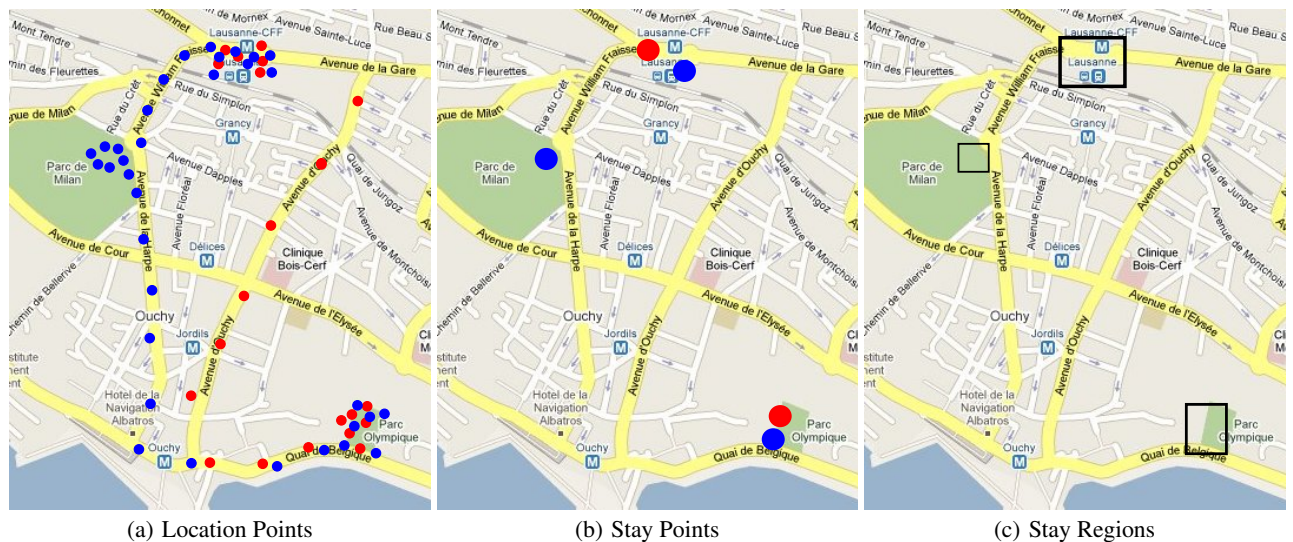


Figure 1: Left: Location points obtained for a hypothetic user during two days (red and blue). Middle: stay points discovered for the two days. Right: stay regions estimated using the previous stay points as input data.

approaches are sufficient to identify indoor places that are smaller than a certain size (e.g. a home), but do not account for outdoor places and larger indoor places (an office complex or a convention center), and are prone to generating false positives (caused by the many possible outdoor GPS shadows).

The previous algorithms depend on properties of the GPS satellite signals to work properly. To avoid this dependence, Kang et al. designed a new algorithm using temporal point clustering [6]. It takes as input a stream of timestamped coordinates derived from a location estimation system, and performs the segmentation and merging steps simultaneously using time-based clustering. One of the main differences from previous works is that location information is obtained using the PlaceLab system [9], which allows the devices to locate themselves by finding radio beacons such as Wifi APs, GSM cell phone towers, and fixed bluetooth devices that already exist in the environment. The main advantage of using this system instead of real GPS data is that it can obtain location information in many places where GPS cannot, and so, more location points can be obtained. Therefore, algorithms can estimate more accurately the places of interest. The main drawback is that PlaceLab has a critical dependence on the availability of beacon locations, otherwise it is impossible to estimate the location of the user. Although there are some public and user-created databases, there are still a lot of beacons without location information in practice, for instance, most of the ones corresponding to the geographical region where our experiments were conducted. Ye et. al. [15] recently presented a similar algorithm that also uses temporal-based clustering, but in this case, location data is obtained using GPS sensors. The main difference (in the strategy to obtain places of interest) is that this algorithm works offline where the one presented in [6] works online.

2.2 Fingerprint-based methods

Fingerprint algorithms, in contrast, obtain a list of places where the user goes, but provide no direct information about where the place is geographically located. Fingerprint-based techniques detect stable radio environments that indicate a stay. They define the fingerprint of a place as a vector of currently visible cell towers or Wifi APs, and use it to recognize when the device returns to a

place. Two of the most important fingerprint-based algorithms are *BeaconPrint* [5] and *PlaceSense* [7]. One of the biggest challenges is dealing with intermittent beacons. According to [7], *PlaceSense* works better in this case than *BeaconPrint* by using multiple successive scans and also thanks to a more robust technique that uses separate mechanisms for detecting place entrances and departures.

Both papers performed a comparative study of the accuracy of their algorithms against geometry-based techniques. In both cases, fingerprint-based algorithms obtained better accuracy results on learning and recognizing places, mainly because geometry-based methods use data from GPS sensors while fingerprint-based use data from GSM and Wifi sensors, who have more coverage in cities. However, the main drawback of fingerprinting is that the exact location information cannot be obtained using this kind of techniques. Then, for many applications where the location of the place must be known, geometry-based are the only choice.

3. AN OVERVIEW OF OUR APPROACH

The algorithm to learn places of interest strongly depends on the data that can be obtained from the sensor. Most systems in the literature for human activity sensing have used custom-made sensor architectures rather than the integrated frameworks existing on phones [11, 1, 6, 15]. However, the need to carry additional devices and their social acceptability limits the use of some of these architectures at large-scale.

In this work, Nokia N95 8GB smartphones have been used to collect location data. Figure 2 shows the framework of the proposed approach. Data has been collected using a client-server system which records GSM, GPS, Bluetooth, Wifi, motion, as well as most of the smartphone applications, in conditions that are practically feasible, with respect to phone usage and battery consumption, for large-scale data collection. One of the main advantages of this system is the design of a set of operation modes (in the client side) that imposes a tradeoff between sensing sampling rate and battery consumption, resulting in efficient sampling at reasonable computational cost. The system consists of a client program that runs on the phone to collect and transmit data over Internet, and a secure

back-end database server to store data. The client stores data until a Wifi connection is available to upload the logged data to the server, which usually happens at home or at work.

From the raw data, location data is extracted (see Section 4). Then the algorithm to estimate stay points is performed using the data obtained for each day of a user. Using the stay points discovered for the user for all the days, the algorithm to discover stay regions is executed. Both algorithms are explained in Section 5. Once places of interest have been extracted, they can be used in many applications as the ones discussed in Section 1.

4. OBTAINING LOCATION POINTS FROM MULTIMODAL MOBILE PHONE DATA

While the extraction of location data directly from GPS is conceptually attractive, it has two practical limitations when using mobile phones. First, GPS is expensive in terms of battery consumption. Second, the GPS sensor on phones fail in practice more often than not, even when using assisted GPS. At the same time, dealing with real-life human location has two inherent advantages. On one hand, most people are habitual and tend to spend a significant proportion of time in exactly the same places [2, 4]. On the other, smartphones are endowed with multiple sensors from which location can be inferred.

Our work relies on a location estimation system that addresses the above four points, recently described in [8]. This system tackles the critical problem of battery consumption by defining an adaptive sensing strategy for location estimation, which integrates the basic knowledge about human habits with the use of multiple phone sensors. More concretely, the phone sensing strategy is defined by a set of *operating modes*, each of which decides what sensors ought to be activated, at what sampling rate, and how to make transitions between each other.

The knowledge about habitual behavior is represented as follows [8]. For each user, the phone periodically scans for Wifi APs to build a map of georeferenced Wifi APs that are observed for a reasonable time period. Urban areas in the developed world often have a high density of Wifi APs in most neighborhoods, so the phone client can build a map of the user's habitual locations, and use this knowledge to avoid activating the GPS sensor unnecessarily. Each phone learns its own Wifi AP location map using the GPS data extracted in the time periods when the phone observes a Wifi, if available. After that, the Wifi AP location is estimated as the centroid of all GPS observations obtained for each AP. From then on, each time the phone observes a Wifi in its map, it switches off the GPS sensor and assigns the corresponding learned location, thus saving battery. This operating mode is called *Wifi map*. This method has similarities to PlaceLab [9], as both try to obtain additional location points to those that the sole GPS sensor can provide by using Wifi APs. The system we use, however, has the important advantage of not needing an external database of beacon locations, but rather learns it by itself. This is important since there are still a lot of beacons without available location information at the geographical region where our experiments were conducted. The system we use also bears similarities with the sensing system reported in [14], which is aimed at recognizing user activities.

The phone also detects when it is *static* by combining the accelerometer, GSM, and Wifi sensors [8]. In this mode, GPS is also switched off to save battery. The phone enters this mode when no significant motion is detected using the accelerometer and the phone re-

mains connected to the same GSM tower or Wifi AP for some time. Similarly to the Wifi location map creation, static location can be estimated using the previous GPS data obtained (if available) just before entering into this mode.

Summarizing, with the multimodal system, there are four possible situations regarding location estimation:

1. In *Wifi Map* mode, location is obtained from the learned Wifi location map.
2. In *Static* mode, location can be obtained from the location obtained just previous to enter such mode (if available).
3. If the phone is in neither of the previous modes, location is directly obtained from GPS when the latter is active and working.
4. If the phone is in neither of the previous modes, and GPS fails (due to no coverage, phone switched off, etc), it is not possible to obtain location.

In real life, the last situation is still common. Therefore, algorithms to learn places of interest should handle this issue.

Regarding energy consumption, the client running on a N95 results in significant savings of battery life, and allows for the use of the phone on a daily basis in regular conditions. Based on an empirical evaluation, three conditions can be observed [8]:

- In the case where the set of *operating modes* is not used, and all sensors are on, the phone is operational for less than three hours before the battery drains.
- In contrast, if the phone stays constantly in the *Wifi Map* operating mode, which is one of the most common ones in practice (occurring in average on 40% of the time), the phone battery lasts for over 20 hours.
- Finally, if the phone stays without interruptions in outdoors (i.e. with Wifi and GPS sensors active), the battery drains in about 8 hours.

In practice, the phone operates between the battery duration bounds described in the last two cases, and can in principle last for a "normal day" for most people, only requiring to be plugged in for recharging once a day, typically when people return home in the evening.

5. ALGORITHMS TO DISCOVER PLACES OF INTEREST

For each day and for each user, the user mobile phone provides a list of consecutive location points $l_p = (p_1, p_2, \dots, p_N)$ where N is the number of location points. Each p_i is defined using a 3-tuple: $p_i = (lat, long, T)$ containing the location (latitude and longitude) and the time when the sensor obtained this location. Note that $\forall i \in [2, N]: T_i > T_{i-1}$.

As explained previously (see Section 4), location points could be obtained from the GPS sensor, from the *Wifi map* mode or from the *static* mode.

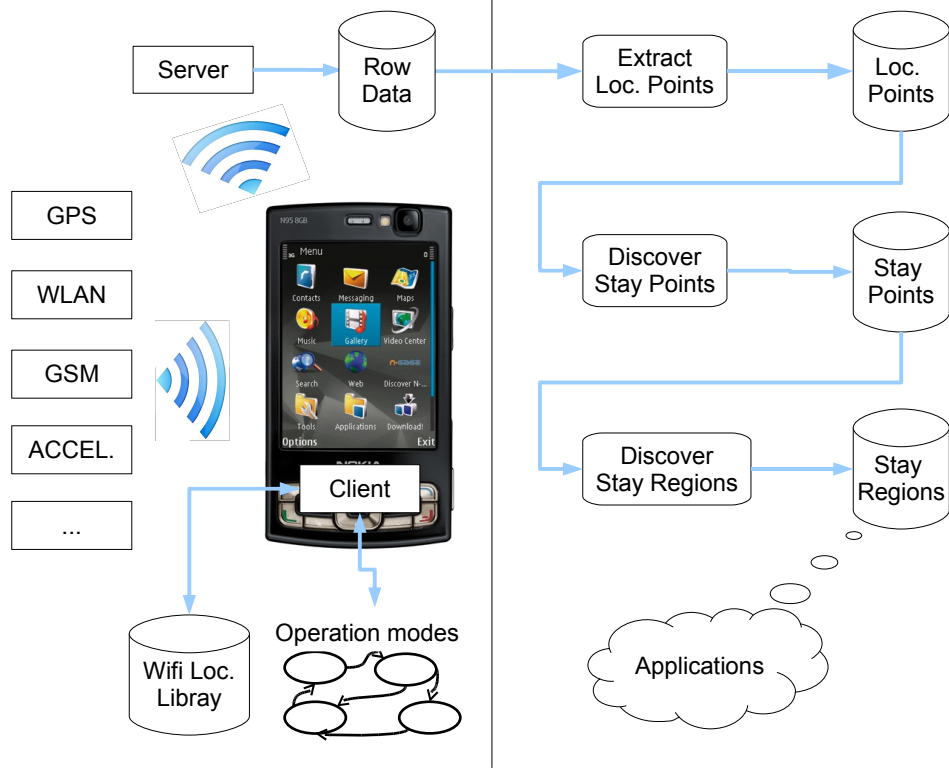


Figure 2: Framework of our approach.

5.1 Estimating stay points from location points

Using the list of consecutive location points l_p for a user and for a day, the objective of the algorithm is to obtain a list of stay points $l_{sp} = (sp_1, sp_2, \dots, sp_M)$, where M is the number of resulting stay points. Each sp_j is defined using a 4-tuple: $sp_j = (lat, long, T^{start}, T^{end})$ containing the location and time when the stay point started and ended. Note that $\forall j \in [2, M - 1]$: $sp_j.T^{start} > sp_{j-1}.T^{end}$ and $sp_j.T^{end} < sp_{j+1}.T^{start}$.

The proposed algorithm to estimate stay points from location points is an improved extension to the one proposed by Ye et. al. in [15]. In this algorithm, a stay point represents a geographic region in which a user stays for a while. More formally, there exists a stay point from location point p_s to location point p_e if the two following constraints are fulfilled:

$$SpaceDistance(p_s, p_e) < D_{max} \quad (1)$$

$$TimeDifference(p_s, p_e) > T_{min} \quad (2)$$

where D_{max} and T_{min} are two tuning parameters. D_{max} is the maximum distance that a user can cover in a place to be considered as stay point. T_{min} is the minimum time that the user must be in the same place to be considered as stay point.

In the proposed algorithm, a new constraint is added:

$$TimeDifference(p_k, p_{k+1}) < T_{max}, \forall k \in [s, e - 1] \quad (3)$$

This new constraint establishes that between two consecutive location points the time difference must be bounded by T_{max} . Then, all the consecutive location points belonging to a stay point must be close in time. The proposed steps to estimate stay points from location points are resumed in Algorithm 1.

Figure 3a illustrates the behavior of the algorithm showing a sequence of location points in a latitude-longitude coordinate space. Two consecutive location points are connected with lines. The line is green when the time difference between the two connected location points is less than T_{max} and it is red otherwise. A stay point is detected (grouping p_1, p_2, p_3 and p_4) since we are assuming that the time difference between p_1 and p_4 is bigger than T_{min} and the physical distance between p_1 and p_4 is less than D_{max} . But, even assuming that the time difference between p_7 and p_{10} is bigger than T_{min} and the physical distance is less than D_{max} , there is not a stay point in the set p_7, p_8, p_9 and p_{10} , since the time difference between p_8 and p_9 is bigger than T_{max} , exceeding the maximum time allowed between two consecutive location points.

When working with real-life data, it is possible to often have situations with substantial time difference between two consecutive location points, which is mainly due to the limitation of accurately sensing location in many real places. One possibility is the one shown in Figure 3b that illustrates the *real* behavior of the user. Between location points p_8 and p_9 , the user visited many other locations (from $p_{8,1}$ to $p_{8,4}$), but they could not be captured by the sensor. Note that without using the third constraint (see Eq. 3) an incorrect stay point could have been obtained between location points p_7 and p_{10} .

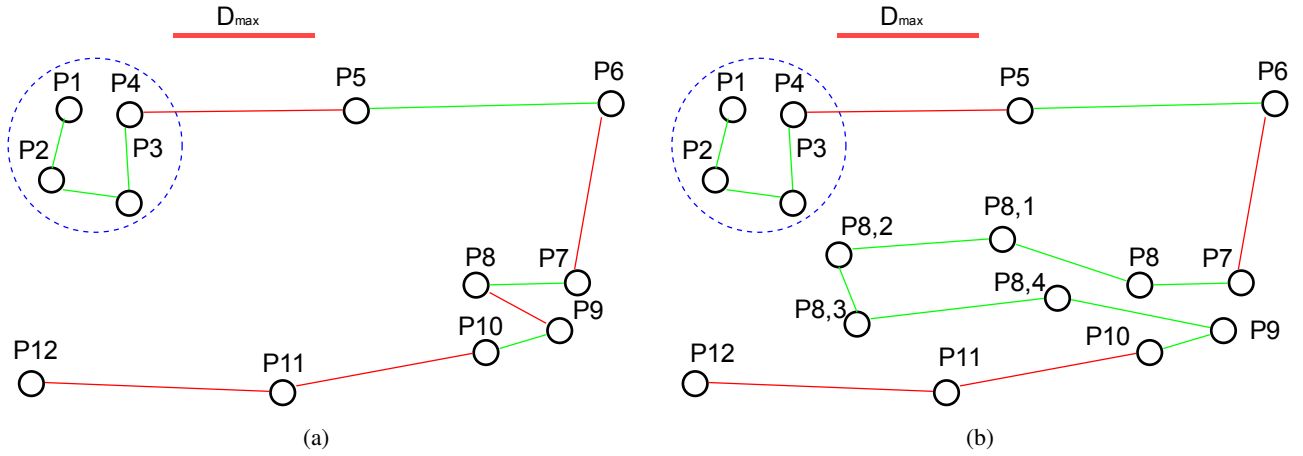


Figure 3: a) An example of the behavior of the proposed technique to estimate stay points from location points. b) An example of which can be the real situation between location points p_8 and p_9 (see text for details).

5.2 Estimating stay regions from stay points

For each user and for each day, the algorithm for estimating stay points is executed. The process of estimating stay regions consists of performing a clustering technique using all the stay points extracted for a user. The general steps to estimate stay regions from stay points for a user are summarized in Algorithm 2. Note that we are assuming that we are working with data from Q days, and therefore there are Q different list of stay points l_{sp} , one for each day.

Some of the clustering techniques that have been used for similar problems in the past include k-means or variants as in [1], density-based as in [15] and grid-based as in [16]. Density-based clustering produces better results than k-means but has the problem of not constraining the clusters size. To deal with this problem, Zheng et. al designed in [16] a new grid-based clustering algorithm which constrains the cluster size. From the results obtained in the experiments shown in Section 6, the grid-based is a better option to perform this task.

6. EVALUATION AND DISCUSSION

6.1 Data acquisition

Our data has been collected from 8 volunteers using the N95 smartphone during 5 continuous months of real life. Users live in three different European communities ranging from small village to middle-size city. Users used the mobile phones in a normal manner, since it was their real (and unique) phone. No instructions were given to the users about which places to visit or about how long they should have stayed in such places. The only recommendation was to carry the phone as frequently as they could, something they would naturally do as it was their real phone.

Instead of having to recharge the phone several times a day (as in [7]) users only needed to recharge once a day (during the night). In total, more than 550,000 location points have been obtained served from 24/7 phone data.

6.2 Evaluation method and collecting the ground truth

To evaluate the effectiveness of the proposed framework, we must first decide on a criterion for evaluation. Intuitively, a place extraction scheme should be judged on how well it identifies the places that a user claims as important. Therefore, it is necessary to have a ground truth of the places reflecting the real-life patterns of users.

Three are the possibilities for collecting ground truth from users:

1. Users can follow a scripted tour of places to stay [7].
2. Users can keep a diary of the names and times when they entered and left every place during the data collection time [7, 5].
3. Users can fill, at the end of the collection period, a survey of the places where they stayed.

The first two options are uncomfortable for users when collecting data during a long period of time. Furthermore, the scripting option might not be looking at truly interesting places from the users viewpoint, or be unrealistic (e.g. requesting people to visit as many places as they can). In addition, the results highly depend on the involvement of each user with the scripted task. The third option above has the main drawback that some places might not be remembered after some time. In our work, to collect ground truth, users filled a survey with the places where they remembered to have stayed for more than 30 minutes.

To evaluate our real-life place-of-interest discovery method, we used an evaluation system partially based on the one recently proposed in [7]. Stay regions (i.e. places of interest) extracted using the proposed framework are called *Discovered*, and places remembered by users are called *Remembered*. Remembered places that have not been discovered are called *Missed*, while places that are both remembered and discovered are called: *Correct* when a single place reported by the user has been discovered as a single place, *Merged* when two different places reported by the user have been discovered as a single place, or *Divided* when a single place reported by the user has been discovered as multiple places. Places that have been discovered but not remembered are called *Ghost candidate*.

Algorithm 1 Algorithm to estimate stay points from location points

Require: $l_p = (p_1, \dots, p_N)$: List of location points.

T_{min}, T_{max} : Time thresholds.

D_{max} : Distance threshold.

Ensure: l_{sp} : List of resulting stay points.

```

i ← 1;
lsp ← ∅;
while i < N do
  j ← i + 1;
  while j < N do
    t ← TimeDifference(pj, pj-1);
    if (t > Tmax) then
      i ← j;
      break;
    end if
    d ← SpaceDistance(pi, pj);
    if d > Dmax then
      t ← TimeDifference(pi, pj-1);
      if t > Tmin then
        [lat, long] ← EstimateCentroid(pk | k ∈ [i, j - 1]);
        Tstart ← pi.T;
        Tend ← pj-1.T;
        sp ← [lat, long, Tstart, Tend];
        lsp ← lsp ∪ sp;
      end if
      i ← j;
      break;
    end if
    j ← j + 1;
  end while
end while

```

If we rely on user memory, all the places discovered but not remembered (i.e. *Ghost candidate* places) are places where the user did not stay. But unfortunately, in practice there are many places where users stayed but that they can not actually remember them. This is a natural limitation of human recall. To deal with this issue, *Ghost candidate* places are divided in *Forgotten* (when the user claims that the place has been correctly discovered *after* reviewing the result) and *False* (in the opposite case).

Finally, it is possible that users remember places where they stayed, but they can not remember if they carried the mobile phone, if it was switched on, etc. Therefore, in principle, some places labeled as *Missed* could (strictly speaking) not be algorithmic errors, since users can remember places that are impossible to estimate since there is no location information available for this moment of the day. Unfortunately, it is impossible to deal with this issue.

With the above definitions, the objective of an automatic method is to obtain more *Correct* and *Forgotten* places, while reducing *False* and *Missed* places. To evaluate the performance, a set of measures (P_α , P_β , R , F_α and F_β), where P , R and F denote *Precision*, *Recall*, and *F-measure*, respectively, have been defined as follows (# stands for "number of"):

Algorithm 2 Algorithm for estimating stay regions from stay points

Require: $l_{sp}^1, \dots, l_{sp}^Q$: set of stay point lists.

Require: D_{max} : Distance threshold.

Ensure: l_{sr} : List of resulting stay regions.

```

SP ← ∅;
for d ← 1 : Q do
  SP ← SP ∪ lspd;
end for
lsr ← Clustering(SP, Dmax);

```

$$\begin{aligned}
P_\alpha &= \frac{\#Correct}{\#Discovered} \\
P_\beta &= \frac{(\#Correct + \#Forgotten)}{\#Discovered} \\
R &= \frac{\#Correct}{\#Remembered} \\
F_\alpha &= 2 * \frac{(P_\alpha * R)}{(P_\alpha + R)} \\
F_\beta &= 2 * \frac{(P_\beta * R)}{(P_\beta + R)} \tag{4}
\end{aligned}$$

Note that, while P_α and F_α are stricter measures, P_β and F_β introduce a correction factor for user forgetfulness.

6.3 Experiments

Several experiments have been performed to evaluate the performance of the proposed approach:

- **Experiment 1** shows that with the combined use of the system explained in Section 3 and our data management strategy (see Section 4), more location points are obtained with respect to using only the GPS sensor.
- **Experiment 2** provides a comparison between two clustering techniques to obtain stay regions from stay points. We demonstrate that the grid-based technique obtains better results than the density-based one.
- **Experiment 3** provides a comparison of our stay points learning strategy with the ones presented by Kang et. al [6] and by Ye et. al. [15]. We demonstrate that on our real-life dataset, our proposed approach outperforms [6] and [15] discovering the users' most important places.

6.3.1 Experiment 1: Extracting location points

To assess the effectiveness of the system for sensing location, we first divided each day of the 5-months period in slots of 5 minutes. Then, for each slot, we counted the number of location points from each source, i.e. if the location point had been obtained directly by the GPS sensor, by the *Wifi map* mode or by the *static* mode. The source with the maximum number of entries was the one that was assigned to this time slot. If there were not location points in a time slot, it was labelled as "No location". This process was repeated for all days and for all users.

Table 1 shows the average values for all users and for all days. Note how by using the operation mode-based sensing system, approximately for 63% of the day it is possible to estimate the location of a

Table 1: Percentage of time when user location data was obtained from GPS, Wifi map mode, static mode, or not obtained.

GPS	Wifi map	static	No location
4%	35%	24%	37%

user and only 4% from the direct GPS sensor. This is an interesting result, as some studies [9, 7] show that GPS coverage is available only 5 – 30% of the time on average for devices carried by users during a typical day.

6.3.2 Experiment 2: Density-based vs Grid-based for stay points clustering

In this experiment a density-based clustering technique is compared against a grid-based one for clustering stay points into stay regions. First, stay points for all users and for all days have been estimated using the proposed approach, setting $T_{min} = 30$ min., $D_{max} = 250$ m. and $T_{max} = 10$ min. Then, both clustering techniques have been applied to estimate places of interest.

DBSCAN [3] has been selected as a representative density-based clustering technique. This algorithm has two main parameters, the minimum number of points in a cluster and the maximum distance between two points belonging to the same cluster. The minimum number of points has been set to 2 and the maximum distance to D_{max} . *DBSCAN* also detects outliers, that in our case are isolated stay points. For each outlier detected, a cluster with only the isolate stay point has been created.

The grid-based technique is the one presented in [16]. The only parameter of this algorithm is the maximum size allowed for a cluster. We used D_{max} for this purpose. Note that in the grid-based method the maximum size of a cluster is bounded by D_{max} (the same value used for discovering places), while in the density-based one, the clusters size is not bounded for any quantity (even if smaller values than D_{max} are used as maximum distance between two points).

Figure 4 shows the typical results of both clustering techniques for a particular user. It is clear that the grid-based technique obtains better results since *DBSCAN* tends to merge stay points with different semantic meaning in the same clusters. In particular, using our real-life dataset, *DBSCAN* has obtained 67 *Correct* and 26 *Merged* places, while the grid-based has obtained 89 *Correct* and only 4 *Merged*.

6.3.3 Experiment 3: Comparative results on place of interest discovering

We compare the proposed approach with Kang et. al. [6] and Ye et. al. [15] ones for estimating stay points from location points. An important point of this experiment is that all algorithms have been tested in the same conditions regarding input location data, since our objective is to test which strategy is the best for learning places from location points. Therefore, the input of three algorithms are the rich location points obtained with our proposed framework instead of just GPS data. In addition, the same parameters have been used in the three algorithms, i.e. $T_{min} = 30$ min., $D_{max} = 250$ m. and $T_{max} = 10$ min. All algorithms use T_{min} and D_{max} , but only ours uses T_{max} .

After the stay points have been extracted using the three algorithms, the grid-based clustering technique has been applied, using each

set of extracted stay points to obtain the places of interest. Table 2 shows the results obtained for the three algorithms.

The proposed technique discovered less stay regions than [15] due to the use of the new constraint (see Equation 3). However, the number of places considered as *Correct* and *Missed* are very similar. Therefore, if we rely only in user remembered places (i.e. considering all *Ghost candidate* places as *False*) the proposed technique obtains better results according to the F_α measure. Note that, while human recall is imperfect, users tend to remember the most important places in their normal lives and therefore our method seems to be more effective at extracting the *most relevant* places for a person.

On the other hand, as discussed previously, users forget a lot of places that they visit but cannot remember when they fill a survey. To know how many *Ghost candidate* places could be considered as *Forgotten*, we showed the results to the users and asked them about the number of *Forgotten* places. In this cases the performance of the three algorithms is quite similar, [15] being slightly better than the proposed one.

6.4 Parameters setting

The algorithms proposed in this work, have only three parameters: D_{max} , T_{min} and T_{max} .

D_{max} is the maximum distance that a user can cover in a place to be considered as a stay point (see section 5.1) and also is the maximum size of a stay region (see section 5.2). On the one hand, big values of D_{max} could merge several places in only one. On the other side, small values could divide a place. The correct value depends on the application. From our experience, values from 200 to 300 meters produce the best results.

T_{min} is the minimum time that the user must be in the same place to be considered as a stay point. High values allow to discover places where the user stayed for long time (home, work, etc.). On the other hand, small values allow to discover places where the user stayed for a few minutes (bus stop, train station, etc.). Similar to D_{max} , the correct value depends on the application. From our experience, values from 20 to 40 minutes produce the best results.

Finally, between two consecutive location points the time difference must be bounded by T_{max} (see section 5.1). High values of T_{max} allow to discover more places, but more *False* places could be discovered too. This could be useful in some applications where we are interested in discovering a lot of places of interest. On the other hand, by setting T_{max} to a low value, only the most significant places for users are going to be discovered, which can be also useful to another application where we are only interested in the most meaningful places for users. The optimum value for T_{max} thus, also depends on the application.

7. CONCLUSIONS

In this paper, a new framework to discover places-of-interest from multimodal mobile phone data has been presented. Mobile phones have been used as sensors to obtain location information from users' real lives. To obtain richer user location points a client system has been installed in the mobile phones, which is able to obtain location information by using GPS, Wifi, GSM and accelerometer sensors. Thanks to the use of this system, it is possible to obtain location data for 63% (approximately) of the day in real life (i.e. not in scripted experiments or artificially encourage mobility). Location

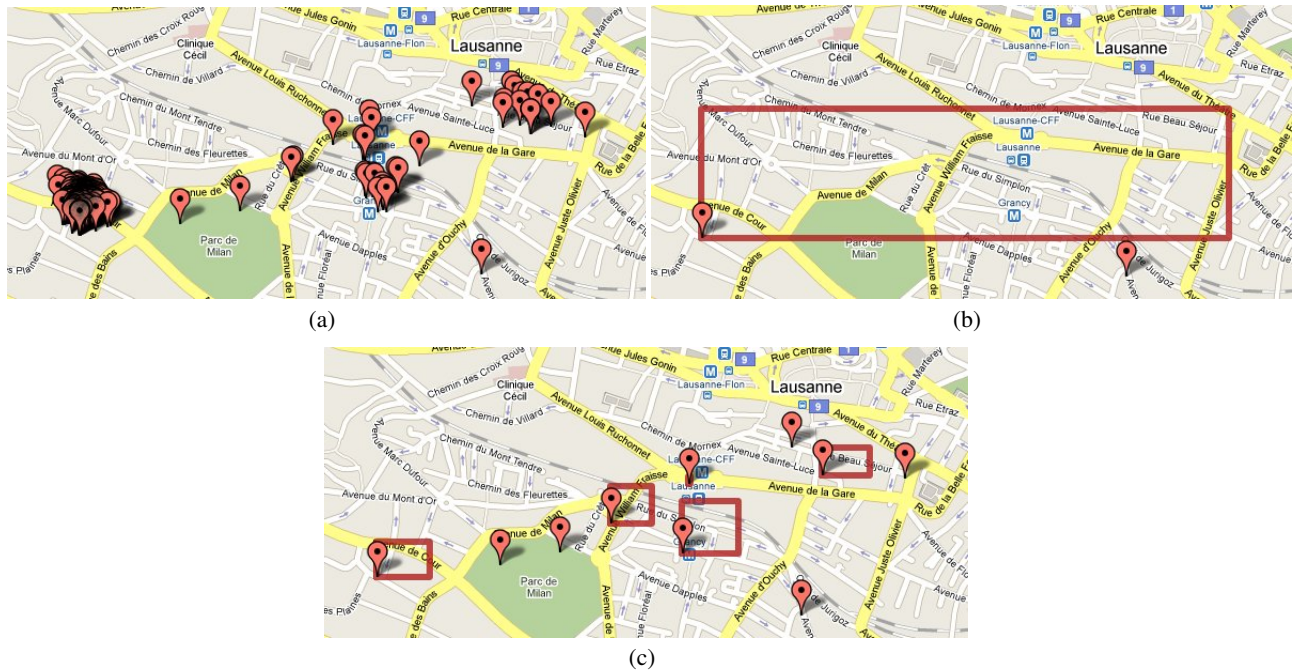


Figure 4: a) Stay points. b) Stay regions obtained using a density-based clustering algorithm. c) Stay regions obtained using a grid-based clustering algorithm.

Table 2: Results obtained using as input the location points obtained for all users and all days.

	Disc.	Remem.	Correct	Missed	False	Forg.	Merg.	Div.	R	P_α	F_α	P_β	F_β
[6]	290	125	88	33	42	161	4	0	0.70	0.30	0.42	0.86	0.77
[15]	314	125	96	25	46	172	4	0	0.76	0.31	0.44	0.85	0.81
Proposed	212	125	89	31	30	92	4	0	0.71	0.42	0.53	0.85	0.78

data are first clustered in stay points by using a time-based method which allows to discover the most significant places remembered by users thanks to the use of a constraint that avoids large time periods between two consecutive location points. Then, stay points are clustered into stay regions (places of interest) using a grid-based clustering technique.

The experiments performed in this work have demonstrated that the proposed framework can obtain more location points corresponding to actual life of people than using only the GPS sensor and without the necessity of having a beacon location database. Our evaluation used 24/7 continuous data over 5 months for 8 people. In addition, we compared our proposed method and two state-of-art techniques (using the same input location points) obtaining better results for discovering the most significant places of interest (according to the users themselves) and similar results discovering other places less important for users.

Future work will focus on implementing the place-of-interest learning technique in the client part of the sensing system of our framework. In this way, when the system discovers a new place, the system could ask to the user about labeling the place to add semantic meaning to the places discovered. In addition, when the user reaches a learned place, the device can ask the user whether he/she is really in this place, improving and simplifying the places-of-interest evaluation procedure.

8. ACKNOWLEDGMENTS

We thank Niko Kiukkonen (NRC) and Olivier Bornet (Idiap) for their contribution to data collection, Jan Blom and Juha Laurila

(NRC) for discussions, and all the volunteers for their participation.

9. REFERENCES

- [1] D. Ashbrook and T. Starner. Using gps to learn significant locations and predict movement across multiple users. *Personal Ubiquitous Computing*, 7, 2003.
- [2] N. Eagle and A. S. Pentland. Eigenbehaviors: identifying structure in routine. *Behavioral Ecology and Sociobiology*, 63(7), 2009.
- [3] M. Ester, H. P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proc. KDD*, 1996.
- [4] K. Farrahi and D. Gatica-Perez. Discovering human routines from cell phone data with topic models. In *Proc. ISWC*, 2008.
- [5] J. Hightower, S. Consolvo, A. LaMarca, I. Smith, and J. Hughes. Learning and recognizing the places we go. In *Proc. Ubicomp*, 2005.
- [6] J. H. Kang, W. Welbourne, B. Stewart, and G. Borriello. Extracting places from traces of locations. *SIGMOBILE Mob. Comput. Commun. Rev.*, 9(3), 2005.
- [7] D. H. Kim, J. Hightower, R. Govindan, and D. Estrin. Discovering semantically meaningful places from pervasive rf-beacons. In *Proc. Ubicomp*, 2009.
- [8] N. Kiukkonen, J. Blom, O. Dousse, D. Gatica-Perez, and J. Laurila. Towards rich mobile phone datasets: Lausanne data collection campaign. In *ACM Int. Conf. on Pervasive Services (ICPS)*, Berlin, Jul. 2010, 2010.
- [9] A. LaMarca, Y. Chawathe, S. Consolvo, J. Hightower, I. E.

- Smith, J. Scott, T. Sohn, J. Howard, J. Hughes, F. Potter, J. Tabert, P. Powledge, G. Borriello, and B. N. Schilit. Place lab: Device positioning using radio beacons in the wild. In *Pervasive*, 2005.
- [10] L. Liao, D. Fox, and H. Kautz. Extracting places and activities from gps traces using hierarchical conditional random fields. *International Journal of Robotics Research*, 26(1), 2007.
- [11] N. Marmasse and C. Schmandt. Location-aware information delivery with commotion. In *Proc. Huc*, 2000.
- [12] D. J. Patterson, L. Liao, D. Fox, and H. A. Kautz. Inferring high-level behavior from low-level sensors. In *Proc. Ubicomp*, 2003.
- [13] S. Reddy, J. Burke, D. Estrin, M. Hansen, and M. Srivastava. Determining transportation mode on mobile phones. In *Proc. ISWC*, 2008.
- [14] Y. Wang, J. Lin, M. Annavaram, Q. A. Jacobson, J. Hong, B. Krishnamachari, and N. Sadeh. A framework of energy efficient mobile sensing for automatic user state recognition. In *Proc. MobiSys*, 2009.
- [15] Y. Ye, Y. Zheng, Y. Chen, J. Feng, and X. Xie. Mining individual life pattern based on location history. In *Proc. MDM*, 2009.
- [16] V. W. Zheng, Y. Zheng, X. Xie, and Q. Yang. Collaborative location and activity recommendations with gps history data. In *Proc. WWW*, 2010.
- [17] Y. Zheng, Q. Li, Y. Chen, X. Xie, and W.-Y. Ma. Understanding mobility based on gps data. In *Proc. Ubicomp*, 2008.
- [18] Y. Zheng, L. Liu, L. Wang, and X. Xie. Learning transportation mode from raw gps data for geographic applications on the web. In *Proc. WWW*, 2008.