Identifying meaningful locations

Petteri Nurmi*, Johan Koolwaaij†
* Helsinki Institute for Information Technology HIIT
P.O. Box 68, University of Helsinki, FI-00014, Finland
petteri.nurmi@cs.helsinki.fi
† Telematica Instituut
P.O. Box 589, NL-7500 AN
Enschede, the Netherlands johan.koolwaaij@telin.nl

***Abstract*— Existing context-aware mobile applications often rely on location information. However, raw location data such as GPS coordinates or GSM cell identifiers are usually meaningless to the user and, as a consequence, researchers have proposed different methods for inferring so-called places from raw data. The places are locations that carry some meaning to user and to which the user can potentially attach some (meaningful) semantics. Examples of places include home, work and airport. A lack in existing work is that the labeling has been done in an ad hoc fashion and no motivation has been given for why places would be interesting to the user. As our first contribution we use social identity theory to motivate why some locations really are significant to the user. We also discuss what potential uses for location information social identity theory implies. Another flaw in the existing work is that most of the proposed methods are not suited to realistic mobile settings as they rely on the availability of GPS information. As our second contribution we consider a more realistic setting where the information consists of GSM cell transitions that are enriched with GPS information whenever a GPS device is available. We present four different algorithms for this problem and compare them using real data gathered throughout Europe. In addition, we analyze the suitability of our algorithms for mobile devices.**

I. Introduction

In context-aware mobile computing, location information has been, without doubt, the most widely studied source of contextual information [1], [3], [16]. The main reason for the situation is that current terminal devices can readily access location related information whereas other sources of contextual information are harder to gather and process. For example, mobile phones can access the GSM cell identifier and PDAs can use information about WiFi access points. In addition, prices of GPS devices with Bluetooth capabilities have decreased significantly and the amount of PDAs that have GPS modules integrated to them has increased rapidly, which makes enriching location information using a GPS a feasible option.

Different sources of location information have their peculiarities. GSM cell tower identifiers give coarse estimates of the location, the numbering of the identifiers is seemingly random and even though operators allow obtaining the current cell identifier they do not offer services to convert cell identifiers into geographic locations. On the other hand, GSM cell tower identifier information is available also indoors, whereas GPS measurements are not. Another weakness in GPS is that, e.g., buildings, trees and glass coatings in cars and trains can cause signal shadowing. Finally, WiFi access point information can be used for positioning only when the exact locations of access points are known (see e.g. [12]).

Regardless of the source of location information, the raw measurements are usually meaningless to the user. As a consequence, much work has been conducted on identifying significant locations, *places*, from the raw data. A place is defined to be a location that is meaningful to the user and to which the user can attach some (meaningful) semantics. For example, home, work and airport are places whereas SomeStreet 42, $(60.42, 42.36)$ or 4287 are not.

A lack in existing work is that, in previous research, locations have been labeled in an ad hoc manner and the motivation for why places would be interesting to a user has not been discussed. To improve the situation, as our first contribution, we use social identity theory to motivate why some locations truly are significant to the user. In addition, we shortly discuss some potential uses for place information implied by social identity theory.

Another flaw in existing work is that most approaches are not suitable for large scale mobile environments. Namely, users seldom have access to GPS information and GSM cell identifiers do not allow separating important places that are near each other, i.e. that are mainly covered by the same cells. Our main contribution is to consider a setting where we log GSM cell transitions and enrich the information using the GPS coordinates of the transition point whenever a GPS device that has a fix is present. To our best knowledge, this kind of setting has not been addressed previously in the literature. We introduce four algorithms for the problem and compare them using data gathered from several users throughout Europe. In addition, we discuss what kind of properties the clusters produced by the algorithms have and how suitable the algorithms are for mobile devices. Our results indicate that the combination of GSM cell transitions and GPS coordinates provides a feasible alternative for continuous data gathering and that it is possible to identify places from this kind of data with high precision.

The rest of the paper is organized as follows: Section II discusses related work, whereas Section III discusses the motivation for inferring significant places and introduces potential uses for place related information. In Section IV we introduce algorithms that can used to cluster GSM cell transitions enriched with GPS information. Section V presents results of applying these algorithms to data we have gathered throughout Europe and, finally, Section VI, concludes the paper and discusses future work.

II. Related work

Existing approaches can be naïvely categorized based on the nature of the used data or based on the type of information that is used to identify the significant locations. In terms of data, the most popular approach has been to use periodically gathered streams of GPS coordinate data. In bounded areas such as office buildings, campuses, research laboratories or individual cities, background information about the physical location of landmark beacons, such as WiFi access points, may be available and hence the second type of data that has been considered is WiFi access point information (access point identifier, signal strength etc.). Finally, GSM cell tower identifiers have been used to identify places [11].

The algorithms for GPS coordinate data typically employ a heuristic approach that is based either on signal loss or on duration. For example, Marmasse et al. [14] first use the geometric distance of succeeding measurements and loss of GPS signal to identify buildings. More specifically, let l_t be the location of user at time t and assume that a signal loss occurs. If the next location l_{t+1} lies within distance of r from the location l_t, the place is inferred to be a building. After the buildings have been identified the number of visits is used to identify significant places.

Ashbrook and Starner [2] use a cut-off parameter $t(\gamma)$ to determine whether the user stays long enough within an area of radius r. If the duration of stay exceeds the value of the cut-off parameter, the location is determined to be a place. A variation of this work is presented by Toyama et al. [19] who use multiple values for the radius parameter r. Initially they use the approach of Ashbrook and Starner to identify locations, after which the value of the radius parameter is decreased and the same procedure is used to identify sub-locations within the previously identified locations. This process is then iterated until no more sub-locations are found.

Kang et al. [6] use background knowledge about the physical location of WiFi access points and the Mac addresses of the access points to identify significant places. The algorithm they use first builds clusters using a customized DBScan ([5]) algorithm, after which a cluster is marked as a significant place, if the user stays long enough within the cluster. A similar approach has also been adopted by Zhou et al. [21], who use a modified DBScan algorithm together with temporal preprocessing for inferring places. The temporal preprocessing ensures that the places are really visited frequently enough and the modification to the DBScan algorithm is needed to cope with signal errors.

The approach by Laasonen et al. [11] works with GSM cell tower identifiers. The used algorithm calculates statistics such as average stay and duration of average visits for recently visited cells. In addition to the statistics, constraints on the graph induced by the GSM cell transitions are used to build cell clusters. Next a duration test is used to check whether the average stay in a location is significant. If this is the case, the cell cluster is considered to be a base, i.e. a significant location.

As concluding remarks, we note that all of the existing approaches first apply a customized version of a density-based clustering algorithm to identify regions of interest, after which temporal constraints are used to detect whether the regions are significant for the user or not.

III. Significant places: Motivation and Uses

In existing work several algorithms have been proposed for the problem of identifying significant locations, but no arguments have been given to support the idea that some places would be significant or interesting in the first place. In this section we attempt to motivate why some places are significant. In addition, we shortly discuss what effect these locations might have to a user's behaviour. Our arguments are based on sociology and, more precisely, on identity theory.

A. Why some locations are significant in the first place?

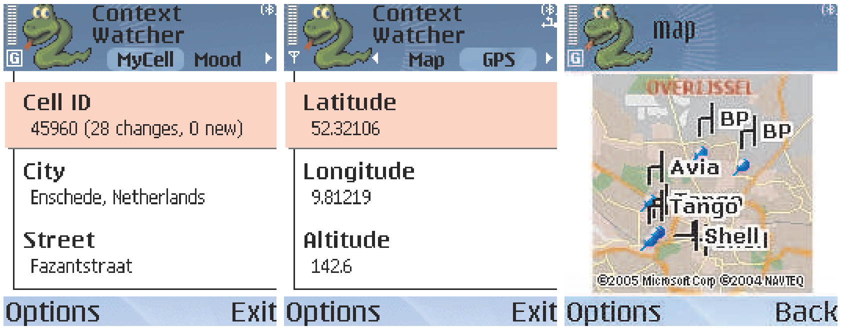
Identity theory is concentrating on studying the relationships between an individual and the society [17], [18]. The central concept is the self of a person, which is seen as comprising of several identities. Each identity, on the other hand, consists of meanings and behavioural expectations that the user associates to specific societal categories (roles) such as father, soccer fan or a democrat. Together with the norms and expectations of the society, the inner meanings and expectations link the persons self to societal categories, which influence the behaviour of the individual.

What is then the link between social identity theory and significant places? The answer is that places where the user spends lots of time typically correspond to specific roles of the user, such as a worker, husband/wife, movie lover etc., and, as a consequence, there is expected to be correlations between the significant locations of a person and his/her behaviour. Hence, significant locations are relevant, if the role of a user depends on the place (not location), in which (s)he currently is. On the other hand, if the user truly acts in different roles in different places, this immediately implies that place information is relevant for personalization.

B. Uses for Place Information: The Context Watcher Case

The fact that place information is relevant for users of context-aware systems should naturally imply some potential uses for place information. To give practical insights into some possible uses, we shortly introduce the Context Watcher application [8], [9] and discuss how Context Watcher utilizes place information. The Context Watcher application is also relevant for later sections as it has been used to gather the data we have used to evaluate our algorithms.

The Context Watcher is a mobile application developed in Python and running on Nokia Series 60 phones. The aim of the Context Watcher is to make it easy for end-users to automatically record, store, and use context information, e.g. for



Cell ID
45960 (28 changes, 0 new)

City
Enschede, Netherlands

Street
Fazantstraat

Latitude
52.32106

Longitude
9.81219

Altitude
142.6

Fig. 1. Screenshots of enriched location data in Context Watcher.

personalization purposes, as input parameter for information services, or for sharing information with family, friends, and colleagues, or even just to log them for future use or to perform statistics on your own life.

In terms of location data, the Context Watcher checks for GSM cell identifier transitions and whenever a cell transition occurs, it sends a snapshot of the current context to a server, which enriches the data. The application has the option to connect a GPS device so that each time a transition occurs, the GPS device is polled for the current location and, if GPS coordinates are available, also those are sent to the server. In addition, if a GPS device has been connected, the application also sends regularly data to the server after around 100 seconds. Thus the data that we consider is far from continuous streams, which have been most widely studied in the literature. On the other hand, we have more information than merely the GSM cell identifiers, which allows us to refine the results of [11].

The Context Watcher itself is a thin application client that interacts with various information services. One of the main information services is the *location provider*, which enriches and refines coarse location information, acts as a repository for location information and exchanges information with authorized parties. For example, the location provider uses GSM cell and GPS latitude-longitude information to deduce the current city and street. For resolving the location from GSM cell identifiers the location provider consults a large cell database [7]. Examples of enriched location information are shown in Fig. 1.

The Context Watcher uses information about significant places in several ways. First of all, with the help of an agenda provider, the Context Watcher creates automatically logs of users daily activities. An example of this is shown in Fig. 2. A related approach is presented in [4] where the concept of entropy is used to construct logs that indicate regularity of daily and weekly activities.

Another use for location information is the use of trajectories. By considering context information from one place (cluster) to another it is possible to recognize user activities such as commuting and, additionally, to discover frequently used routes as has been shown, e.g., in [13], [10].

In terms of roles, time related information can be combined with place information to deduce relevancies of significant places in particular contexts. As a simple example, if the place corresponding to work/office has been identified mainly from data that has been gathered during weekdays, the concept of roles allows us to exclude the office cluster on weekends. As another example, if a worker goes past a church (e.g. when coming from work), it naturally does not mean it is Sunday and the church place is less likely to be relevant than if it truly is Sunday.

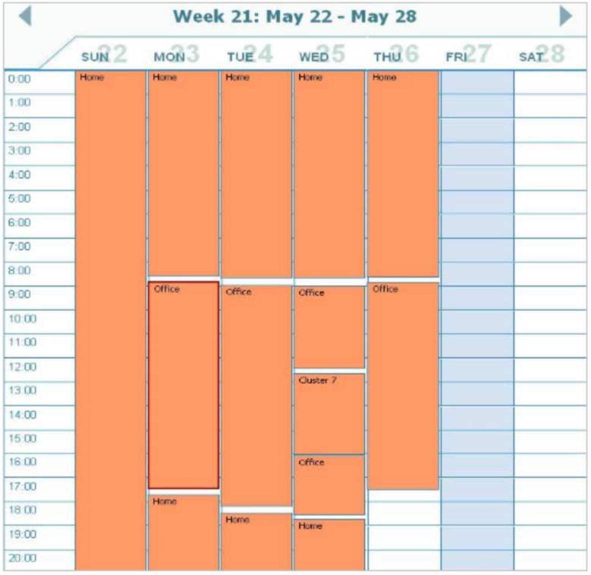


Fig. 2. An example of a user's daily log.

IV. Identifying Places

As we now know that some locations are significant to the user, we can move on the problem of identifying them. In our problem setting, the Context Watcher application (see previous section) monitors for cell changes and whenever a cell change occurs, it sends the available location information to a *Location Provider* that resides on a server. If there is a GPS device connected to Context Watcher, also the GPS coordinates at the transition point are sent to the location provider. Finally, when a GPS device (that has a fix) has been connected, data is sent to the server after around 100 seconds.

If no GPS coordinates are provided, the location provider can use GPS measurements of other users to give an estimate for the GPS coordinates of the cell. All in all, the setting we are considering results in missing data: the GSM cell identifier is almost always present as well as country information. Latitude and longitude information we have approximately in 70% of the cases, city information in 60% of the cases and street information in 50% of the cases. In our setting relatively many people had GPS devices available, but in reality the situation would not be as good.

The location provider is also responsible for identifying places from the available location information and in this section we present four algorithms for identifying places in the described setting. The first two algorithms are based on graph clustering whereas the two last ones are based on duration and cell transition information. Also our algorithms are variants of spatial clustering and, the two last ones use also temporal processing.

A. Heuristic Graph Clustering

The discrete nature of GSM cell identifiers allows us to consider a graph analogue for the problem of identifying places from GSM transition data. The *cell graph* is defined to be a graph whose vertices correspond to GSM cell identifiers and where is an edge between vertices i and j, if the data has a cell transition from cell i to cell j. We furthermore assume that the graph is weighted and use $w_{i,j}$ to denote the weight between GSM cells i and j. Thus the GSM cell transitions induce a (weighted) directed graph, which we use to identify places.

Algorithm 1 Graph clustering algorithm

```
 1: Input: Threshold parameters $\alpha, \beta, \gamma$
 2: Output: List $C_L$ of clusters and their confidences
 3: Initialization: L := $\emptyset$
 4: for each cell $c$ do
 5:    centroid(c) = [0.0,0.0]
 6:    for each location $L$ containing cell $c$ do
 7:       if L.latitude $\neq 0$ and L.longitude $\neq 0$ then
 8:          centroid(c) += [L.latitude,L.longitude]
 9:       end if
10:    end for
11:    centroid(c) = centroid(c) / #L (#L = number of location measurements)
12: end for
13: Remove cells for which centroid(c) = $(0.0, 0.0)$.
14: for each cell $c_i$ do
15:    for each cell $c_j \neq c_i$ do
16:       if count($c_i$) $>$ $\alpha$, count($c_j$) $>$ $\alpha$, exp($\beta$ d(centroid($c_i$),centroid($c_j$)) $< \gamma$ then
17:          add $\{c_i, c_j\}$ to $C_L$
18:       end if
19:    end for
20: end for
```

We consider first a heuristically motivated algorithm that is outlined in Alg. 1. The algorithm consists of two main loops. In the first loop (lines 4 - 12), the centroid of GPS measurements is calculated for each cell. As the next step (line 13), we remove the cells for which no GPS information was available. As was mentioned in the beginning of this section, this does not mean that we require users to carry a GPS device with them. However, for a cell to be considered in the clustering we require that somebody has made at least one GPS measurement while moving from the cell to another.

The second loop of the algorithm (lines 14 - 20) prunes the set of possible clusters by requiring that cells have been seen enough many times. At line 16 the centroids of the different cell pairs are compared and at line 17 a cluster (place) is formed whenever the centroids are close enough. The closeness is measured using exponential decay on the distance and a threshold for the "probability" given by the decay function. To return to the graph analogue, the algorithm uses the Euclidean distance between the centroids of two cells as the weight of the edge that links them and the decay function is used to test whether the weights are sufficiently small.

The main weaknesses of the heuristic graph clustering algorithm are that it is not based on any theoretical framework, which prevents from giving guarantees on its performance, and that the results depends on the values of three tuneable parameters that can be non-intuitive to set. However, as is shown in Section V, the algorithm produces accurate results in realistic settings. A less intuitive fault that we have identified in our experiments is that the clusters begin to clutter over time as more and more cells are merged to the same cluster. A potential solution to overcome this effect is to tune the cut-off threshold or the parameter of the exponential decay according to the size of the cluster.

Although we have presented the algorithm as an offline algorithm that has $\mathcal{O}(|L| + |C|^2)$ time requirements, where $|L|$ is the number of location measurements and $|C|$ is the number of cells, it is easy to convert the method to be more suitable for mobile settings. Namely, the centroid and count statistics can be updated incrementally and, in addition, line 16 can be transformed into an online version, which reduces the time complexity of the algorithm to $\mathcal{O}(|C|)$. In this case the memory requirements of the algorithm are also $\mathcal{O}(|C|)$, which makes the algorithm better suited for mobile devices. The online version of the algorithm is given in Alg. 2

B. Spectral clustering

A natural alternative for the heuristic graph clustering is to use an algorithm that has a better theoretical basis. To this end, we applied spectral clustering to the problem. Spectral clustering is a relatively new clustering technique that is based on spectral graph theory. The idea is to define an objective function for partitioning a graph into k sub-partitions and to minimize an objective function that defines the "goodness" of a cut. In the literature several measures for the goodness of a cut have been formulated and we refer to [20] for a more comprehensive treatment of the topic. The algorithm that

Algorithm 2 Online variant of the graph clustering algorithm

```
Input: Location measurement $l$, List $C_L$ of clusters and their confidences
if coordinates($l$)$\neq$(0.0, 0.0) then
  centroid(cellid($l$)) += coordinates($l$)
  centroidcount(cellid($l$)) += 1
end if
count(cellid($l$)) += 1
for each cell $c \neq$ cellid($l$) do
  if count(c) $> \alpha$ and count(cellid($l$)) $> \alpha$ then
    $v$ = centroidcount(cellid($l$))$^2 *$ centroid(c) - centroidcount(c)$^2 *$ centroid(cellid($l$))
    $w$ = centroidcount(cellid($l$))$^2 *$ centroidcount(c)$^2 * \log(\gamma)/\beta$
    if $v < w$ then
      Add {cellid($l$), c} to $C_L$
    end if
  end if
end for
```

we use in our experiments is presented in [15]. As spectral clustering uses only locality information in the clustering process we have used duration and cell count information as a post-processing step to prune out the irrelevant clusters, i.e. those that are less likely to be places. The algorithm that we have used in our experiments is outlined in Alg. 3.

Algorithm 3 Spectral clustering

```
Input: Data $y_1, \dots, y_n$, Affinity parameter $\sigma$, Magnitude of smallest eigenvector to consider $\gamma$
for Each pair of data $(y_i, y_j)$ do
  $A_{i,j} = \exp\left(-\frac{||y_i - y_j||}{2\sigma^2}\right)$ {Calculate affinity matrix}
end for
for Each index $i = 1, \dots, n$ do
  $D_{i,i} = \sum A(k,:)$ {Form a diagonal matrix $D$ from the affinity matrix $A$}
end for
$L = D^{-1/2} A D^{1/2}$ {Calculate the Laplacian}
$[V, E]$ = eig($L$) {Calculate the eigenvalues and eigenvectors of $L$}
$k = \| \max_i E(i) \| < \gamma$, {Assume the eigenvalues are sorted}
$X = V[1:k]$, {Use the $k$ first eigenvalues}
$Y = X$
for Each index $i = 1, \dots, n$ do
  for Each index $j = 1, \dots, n$ do
    $Y(i,j) = X(i,j)/\sqrt{\sum X(i,:)^2}$
  end for
end for
clusters = kmeans(Y,subsets);
Return: cellids for which count(cellid) $>$ $\text{avg}_{\text{visit}}$ and durationOfStay(cellid) $>$ $\text{avg}_{\text{durationOfStay}}$
```

The strengths of spectral clustering are that the method is based on a theoretically sound framework and that the algorithm can utilize small scale locality information to produce really compact clusters using only coordinate data. However, a major drawback of the algorithm is its applicability to mobile phones. Namely, the memory requirements of the algorithm are $\mathcal{O}(|C|^2)$, where $|C|$ is the number of cells. A potential alternative is naturally to run the clustering overnight on a server and to reduce the number of data points that are considered in a single run.

C. Duration based grid clustering

A third approach makes use of the natural aspect that clusters are easily identified by the amount of time that a user spends in a certain location. A complicating factor is however that the nature of the location data is diverse, which results in strongly varying accuracy and incomplete information in a substantial percentage of the location measurements. More concretely, accuracy typically varies between 5 and 5000 meters and the missing data includes latitude-longitude pairs for cells that could not be resolved by the cell-id database.

Our approach detects peeks in the duration as a function of latitude and longitude, which are discretized into a grid with a certain resolution. The accuracy of a location measurement is used to distribute the duration uniformly over all grid points that have a distance to the estimated location lower than the accuracy. A cluster is then formed by connecting neighboring grid points that have a relative duration larger than a threshold percentage. We apply duration and distance checks that make sure that two consecutive location measurements are actually consecutive in reality and are not caused by missing data, because the user did not run the application for some amount of time. A formal description of the algorithm is given in Alg. 4.

Algorithm 4 Duration-based grid clustering

```
for Each measurement $m_i \in M$ do
  if duration($m_i$) < MAXDURATION AND $d(m_i, m_i + 1)$ < MAXDIST then
    for Each gridpoint $g$ in $G$ with $d(g,$ grid($m_i$) $<$ accuracy($m_i$) do
      duration(g) += duration($m_i$) / count(G)
      totalduration += duration($m_i$)
    end for
    for Each gridpoint $g$ in $G2$ with duration(g) / totalduration > p do
      if $g$ not in any cluster then
        Cluster c = findAllNeighbours(g)
        Clusters.add(c)
      end if
    end for
  end if
end for
```

The clusters are then applied using a simple rule, according to which the user is in cluster c_k if $grid(m_j)$ belongs to c_k.

D. Frequent transitions

The last algorithm we consider is based on observing frequent cell transitions. It is well known that while staying at the exact same location for a longer time, a user might observe frequent changes between 2, 3 or even more covering cells. This algorithm exploits this behavior to find the personal clusters. In this case, each cluster consists of a collection of cells. The clusters are updated by starting from the existing clusters, constructing the personal transition matrix over 1 month time, and collecting all cell pairs (A,B) where both $P_{trans}(A, B)$ and $P_{trans}(B, A)$ are higher than a threshold probability. If only A belongs to a cluster already, B is assigned to the same cluster as A, and vice versa. If both do not belong to a cluster, a new cluster is formed with A and B as cells. And if both already belong to a different cluster, no action is undertaken.

The advantage of this is that it works even for areas where the location of cells is unknown. However, a disadvantage is that the algorithm introduces a network operator dependency, since cross operator cell transitions are not frequent. As a result the user has double clusters when he is abroad (and hence easily switches from operator), and the user has to start from scratch again when he starts a new subscription with another operator in his home country.

V. Experiments

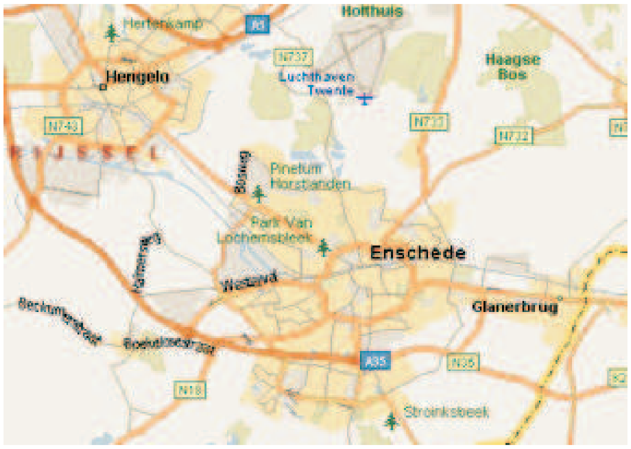


Fig. 3. The area of interest.

To provide insights into the presented algorithms, we have applied the methods for data gathered using the Context Watcher application (see Section III). The overall user base of Context Watcher consists of 65 users of which about 25 are active during a month. However, to better compare the algorithms, we have selected as a common use case data from the Netherlands. A map of the area we are considering is shown in Fig. 3. As the nature of the data in our setting differs from the data that is considered in previous work, existing algorithms were not applicable and thus no comparison to earlier work could be done.

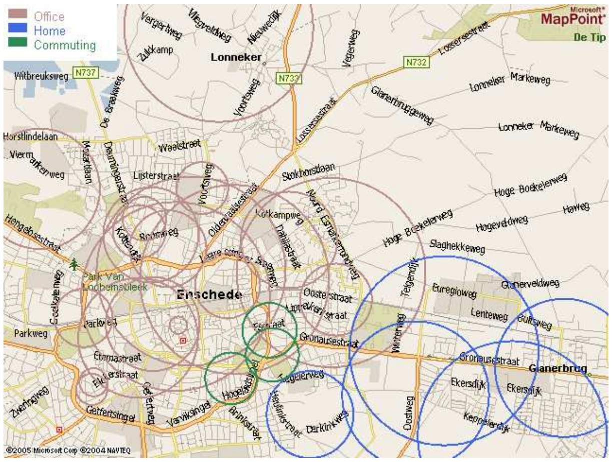


Fig. 4. Results from using the heuristic graph clustering algorithm.

The clusters produced by the heuristic graph clustering algorithm (Alg. 1) are shown in Fig. 4. As can be seen from the figure, the clusters have cluttered and the office cluster contains most of the city centre, which is not the case in reality. Thus the clusters that the algorithm produces are inaccurate and the method is able to give only coarse location information, i.e. it is not suited for identifying places. We have previously used this method in the Context Watcher application and our experience is that when going to a new place the clusters produced by the algorithm are accurate, but once more data is obtained, the cluttering effect starts to occur. In addition, the threshold parameters should be different for densely and for sparsely populated areas, which clearly is a major disadvantage.

The second set of clusters is produced using spectral clustering (Alg. 3). We used the value 0.10 for the affinity parameter and the same value was used also as the cut-off value for the magnitude of smallest eigenvector. The results of the algorithm are shown in Fig. 5.

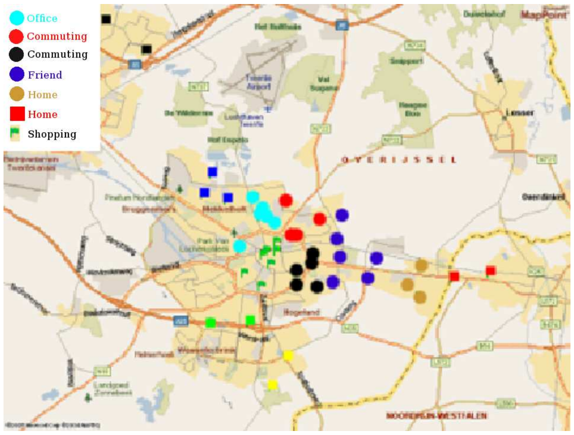


Fig. 5. Results from using spectral clustering. The markers represents means of clusters. There are two home clusters due to the German border, which causes a switch in the operator.

As can be seen from the figure, the results produced by spectral clustering are rather compact and they belong to clearly separate areas. A possible drawback is that the algorithm produces commuting clusters, which might not be significant to the user. However, the discussion whether users find commuting clusters relevant or not is out of scope for the paper. On the other hand, an advantage of the commuting clusters is that they ease route prediction. At the moment we used the average count and average duration for post-processing and a more detailed analysis of durations would allow us to give also significances for the clusters. Finally, an alternative is to merge the commuting clusters with clusters corresponding to places, but at the moment we do not use this approach to avoid cluttering.

The results of the duration based grid clustering are shown in Fig. 6. The results indicate that the algorithm is able to identify correctly true locations, but the figure points out also a potential weakness of the algorithm. Namely, also the grid clustering algorithm produces commuting clusters. In the experiment the home cluster of one of the authors was assigned an overall duration of around two percentage units, which would mean that the home would be easily pruned out if further processing is used. The main reason for this is that the number of grids the algorithm produces is too large for the area it is considering and because the home cluster is not nearby to other areas of interest it gets only a small fraction of the overall duration. Another disadvantage of the grid based clustering algorithm is that it produces multiple grids for places that are covered by large cells. Thus, instead of producing a grid around the office, the office is in the middle of two grids. As a consequence, also the grid clustering requires merging of grids. However, a major advantage of the algorithm is that it can nicely handle data of different accuracies, which we consider to be more important than the smearing of clusters.

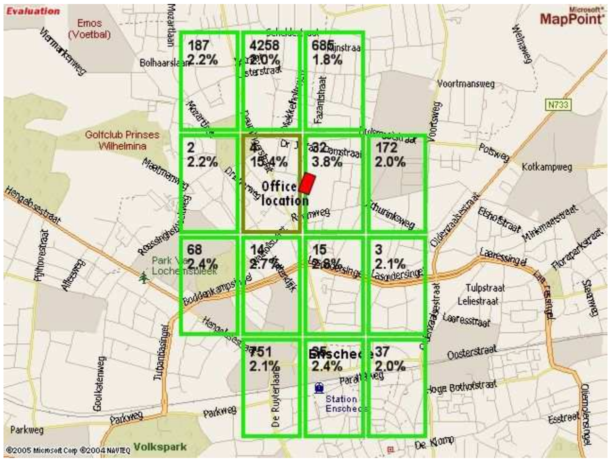


Fig. 6. Results from using the duration based grid clustering.

Finally, the results of the transition based clustering (Alg. 4) are shown in Fig. 7. The results are fairly similar to the results of spectral clustering with the only difference being the church cluster. Thus also the transition based algorithm suffers from amount of commuting clusters.

As concluding remarks, the algorithms that we have used seem to work rather accurately. Some issues such as cluster size and meaningfulness of the clusters, i.e. do they truly correspond to places, are issues that are still requiring further work.

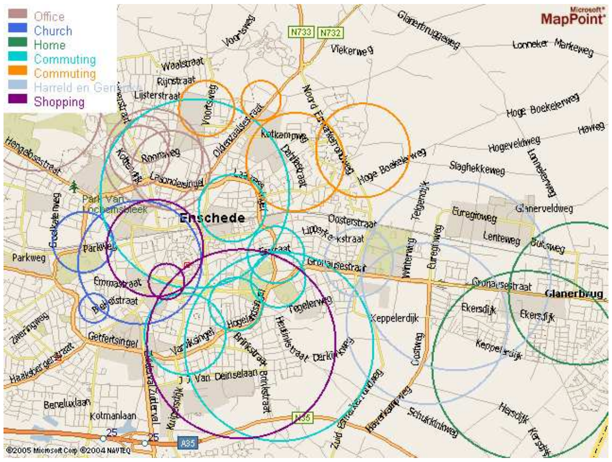


Fig. 7. Results from using the transition based clustering.

VI. Conclusions and future work

In this paper we have given a motivation for why some locations are significant and we have shortly discussed potential uses for place information implied by the arguments used to support our motivation. In addition, we have considered the problem of identifying places from GSM data that is enriched with GPS coordinates whenever a GPS device is available. We presented four algorithms for the problem and compared them using empirical data gathered throughout Europe. In addition, we have analyzed the properties of the clusters the algorithms produce as well as the suitability of the methods for mobile environments. Namely, it is important to have a clustering algorithm that performs well when applied in an iterative manner. Because the caching duration of the location context provider is limited (and in the future even user-controlled), the algorithm needs to start from the existing clusters, observe the measurements over a limited time period, and use these observations to adapt the existing clusters.

In general, we should be able to cope with incomplete data sets. Sometimes GSM cell identifier is the only thing we have, sometimes we can convert identifiers into location with a certain range, sometimes we can convert geolocation into address, but often this fails. So for example, if we want to use e.g. city or geolocation info in our algorithms we should be prepared for 'holes' in the data. In the future our goal is to develop a generative probabilistic model that allows to replace missing values with expected location and accuracy information. Probabilistic clustering techniques can then be used to identify places in this setting.

At the moment we are using an unsupervised approach where the clusters are derived automatically, and the user can only give them a name and a type (link to ontology). In the future, our goal is to consider also semi-supervised approaches that allow more control to the user. This requires however a more intelligent cluster management interface than the one existing in Context Watcher at the moment.

The users of Context Watcher have found the clustering useful and, as an example, when you arrive at a location where a business meeting is held, already on the second day clusters (places) corresponding to hotel and the meeting place are identified. The users then simply name the clusters, after which the cluster names appear automatically in log reports of user's daily activities and as labels for pictures. Although the places that are visited really frequently are harder to handle due to the cluttering effect, our experience is that the clustering works well. To support this argument, the number of places identified from Enschede (150.000 inhabitants) does not exceed 7 although we have gathered data for 9 months almost continuously.

VII. Acknowledgements

This work was supported in part by the European Union Information Society Technology in the Sixth Framework Program under the contract number IST-511607 (MobiLife project). The authors also wish to thank the rest of the Context Watcher development team: Anthony Tarlano, Marko Luther, Agathe Battestini, Raju Vaidya and Bernd Mrohs.

References

[1] G. D. Abowd, C. G. Atkeson, J. Hong, S. Long, R. Kooper, and M. Pinkerton. Cyberguide: A mobile context-aware tour guide. *Wireless Networks*, 3(5):421 – 433, October 1997.

[2] D. Ashbrook and T. Starner. Using GPS to learn significant locations and predict movement across multiple users. *Personal and Ubiquitous Computing*, 7(5):275 – 286, 2003.

[3] G. Chen and D. Kotz. A survey of context-aware mobile computing research. TR2000-381, Dartmouth College, Hanover, NH, USA, 2000.

[4] N. Eagle and A. Pentland. Reality mining: Sensing complex social systems. *Personal and Ubiquitous Computing*, Online first, 2005.

[5] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In E. Simoudis, J. Han, , and U. Fayyad, editors, *Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 226 – 231. AAAI, 1996.

[6] J. Kang, W. Welbourne, B. Stewart, and G. Borriello. Extracting places from traces of locations. In *Proceedings of the 2nd ACM international workshop on Wireless mobile applications and services on WLAN hotspots (WMASH)*, pages 110 – 118. ACM Press, 2004.

[7] J. Koolwaaij. Mapping the GSM landscape. In *Proceedings of SVG open*, 2005.

[8] J. Koolwaaij, A. Tarlano, M. Luther, A. Battestini, P. Nurmi, R. Vaidya, and B. Mrohs. Context Watcher. `http://www.lab.telin.nl/~koolwaaij/showcase/crf/cw.html`, 2005.

[9] J. Koolwaaij, A. Tarlano, M. Luther, P. Nurmi, B. Mrohs, A. Battestini, and R. Vaidya. Sharing context information in everyday life. In *Proceedings of the 4th International Conference on Pervasive Computing PERVASIVE*. Springer-Verlag, 2006. Under review.

[10] K. Laasonen. Clustering and prediction of mobile user routes from cellular data. In *Proceedings of the 9th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD)*, volume 3721 of *Lecture Notes in Artificial Intelligence*, pages 569–576. Springer-Verlag, 2005.

[11] K. Laasonen, M. Raento, and H. Toivonen. Adaptive on-device location recognition. In A. Ferscha and F. Mattern, editors, *Proceedings of the Second International Conference on Pervasive Computing (PERVASIVE)*, LNCS 3001, pages 287–304. Springer, 2004.

[12] A. LaMarca, Y. Chawathe, S. Consolvo, J. Hightower, I. Smith, J. Scott, T. Sohn, J. Howard, J. Hughes, F. Potter, J. Tabert, P. Powledge, G. Borriello, and B. Schilit. Place Lab: Device positioning using radio beacons in the wild. In *Proceedings of the 3rd International Conference on Pervasive Computing PERVASIVE*, volume 3468 of *Lecture Notes in Computer Science*, pages 116 – 133. Springer-Verlag, 2005.

[13] L. Liao, D. Fox, and H. A. Kautz. Learning and inferring transportation routines. In *Proceedings of the Nineteenth National Conference on Artificial Intelligence (AAAI)*, pages 348 – 353, 2004.

[14] N. Marmasse and C. Schmandt. A user-centered location model. *Personal and Ubiquitous Computing*, 6(5 - 6):318 – 321, 2002.

[15] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and algorithm. In T. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Processing Systems*, volume 14, pages 849 – 856, Cambridge, MA, 2002. MIT Press.

[16] B. Rao and L. Minakakis. Evolution of mobile location-based services. *Communications of the ACM*, 46(12):61 – 65, December 2003.

[17] J. Stets and P. Burke. Indentity theory and social identity theory. *Social Psychology Quarterly*, 63(3):224 – 237, 2000.

[18] S. Stryker and P. Burke. The past, present and future of social identity theory. *Social Psychology Quarterly*, 63(4):284 – 297, 2000.

[19] N. Toyama, T. Ota, F. Kato, Y. Toyota, T. Hattori, and T. Hagino. Exploiting multiple radii to learn significant locations. In T. Strang and C. Linnhoff-Popien, editors, *Proceedings of the 1st International Workshop on Location- and Context-Awareness (LoCa)*, volume 3479 of *Lecture Notes in Computer Science*, pages 157 – 168, Berlin Heidelberg, 2005. Springer-Verlag.

[20] D. Verma and M. Meila. A comparison of spectral clustering algorithms. UW-CSE-03-05-01, University of Washington, 2003.

[21] C. Zhou, D. Frankowski, P. Ludford, S. Shekhar, and L. Terveen. Discovering personal gazetteers: an interactive clustering approach. In *Proceedings of the 12th annual ACM international workshop on Geographic information systems (GIS)*, pages 266 – 273, New York, NY, 2004. ACM Press.