Automatic identification of informal social groups and places for geo-social recommendations

Ankur Gupta and Sanil Paul

Department of Computer Science,
New Jersey Institute of Technology,
University Heights, Newark, NJ 07102, USA
E-mail: ag59@njit.edu
E-mail: sp286@njit.edu

Quentin Jones

Department of Information Systems,
New Jersey Institute of Technology,
University Heights, Newark, NJ 07102, USA
E-mail: qjones@njit.edu

Cristian Borcea*

Department of Computer Science,
New Jersey Institute of Technology,
University Heights, Newark, NJ 07102, USA
E-mail: borcea@cs.njit.edu
*Corresponding author

Abstract: Mobile locatable devices can help identify previously unknown ad hoc or semi-permanent groups of people and their meeting places. Newly identified groups or places can be recommended to people to enhance their geo-social experience, while respecting privacy constraints. For instance, new students can learn about popular hangouts on campus or faculty members can learn about groups of students routinely having research discussions. This paper presents a clustering algorithm based on user copresence that identifies such groups and places even when group members participate to only a certain fraction of meetings. Simulation results demonstrate that 90–96% of group members can be identified with negligible false positives when the user meeting attendance is at least 50%. Experimental results using one-month of mobility traces collected from smart phones running Intel's PlaceLab location engine successfully identified all groups that met regularly during that period. Additionally, the group places were identified with good accuracy.

Keywords: mobile social computing; location aware recommender systems; group identification; place identification.

Reference to this paper should be made as follows: Gupta, A., Paul, S., Jones, Q., and Borcea, C. (XXXX) 'Automatic identification of informal social groups and places for geo-social recommendations', *Int. J. Mobile Network Design and Innovation*, Vol. X, No. Y, pp.XXX–XXX.

Biographical notes: Ankur Gupta is a PhD candidate in the Department of Computer Science at the New Jersey Institute of Technology (NJIT). His research interests include mobile and ubiquitous computing, middleware and algorithms. He received an MS in Computer Science from NJIT in 2005.

Sanil Paul is working toward an MS in Computer Science at NJIT. His research interests include ubiquitous computing and location-aware systems.

Quentin Jones is an Assistant Professor at NJIT. He is the Director of NJIT's SmartCampus Project, an effort to explore location-aware community system design, utility and social impacts. His research and teaching focus is social computing with an emphasis on the design of collaborative environments. He has a PhD in Information Systems from Haifa University, Israel.

Cristian Borcea is an Assistant Professor in the Department of Computer Science at the NJIT. His research interests include mobile and ubiquitous computing, ad hoc networks, middleware and distributed systems. He is a Member of ACM, IEEE and Usenix. Cristian received a PhD in Computer Science from Rutgers University in 2004.

1 Introduction

Internet-based social networking applications such as Facebook (2004), MySpace (2003) and LinkedIn (2002) have experienced a huge success during the last few years. Existing location technologies (Bahl and Padmanabhan, 2000; Enge and Misra, 1999; LaMarca et al., 2005; Priyantha et al., 2000), which proliferated on many mobile devices such as smart phones, can be used to build on this success and deliver *location-aware* social computing applications. With research (Jones et al., 2007) showing that users are increasingly willing to share their location in return for services, these applications can provide geo-social recommendations about people, places and events of interests anytime, anywhere. The first steps in this direction have already been taken by a number of context-aware recommendation systems (Espinoza et al., 2001; Heijden et al., 2005; Takeuchi and Sugimoto, 2005; Yang et al., 2008). While these systems consider location or user preferences when making recommendations, they do not take into account group membership and associations between groups and places. If captured and properly used, group membership information can enhance the user profiles, thus improving the quality of people-to-people recommendations (Jones et al., 2004). Similarly, group-place associations can improve the quality of place recommendations by enhancing the semantics of the place with social information. However, identifying social groups and their associated places is a challenging task.

Social groups can be divided as either formal or informal. Formal groups (e.g. students in a class, faculty members of a department) have a formal organisational structure as well as advertised meeting places and times. These groups and their meeting places can easily be identified using web sites, databases, notice boards or mailing lists. On the other hand, informal groups are very hard to identify due to their volatile or semi-permanent nature. Examples of informal groups include a study group for a class, faculty that routinely have lunch together, coworkers who play poker once in a while, or neighbours who go together to the mall on Saturdays. These groups tend to evolve out of collaborating individuals with similar interests, and they are typically unknown to people outside the group. Unlike formal groups, their information (e.g. type, members, meeting places and times) is not registered with an information database or service. However, this information can be used, while respecting privacy constraints, to provide valuable recommendations that improve users' geo-social experience. For example, new students can learn about popular hangouts for social activities on campus or faculty members can learn about groups of students meeting to discuss a certain research topic.

This paper presents Group-Place Identification (GPI), an algorithm for automatic identification of informal social group members and group-place associations using community mobility traces. GPI can be incorporated in different location-aware social computing applications that deliver geo-social recommendations. While users can potentially provide data about informal social groups and places, we believe that an automatic method is much more accurate for two reasons. Firstly, it is possible that only a small fraction of the users will introduce these data manually. And secondly, the information introduced by users can contain errors either by mistake or maliciously. GPI can use mobility traces acquired from any type of location technology. From a user privacy perspective, however, systems that compute the location on the mobile devices (e.g. Enge and Misra, 1999; La Marca et al., 2005) are preferable because they give users control over what parts of mobility trace are shared.

So far, mobility traces have only been used in algorithms that identify significant places for individual users, such as Kang et al. (2004) and Hightower et al. (2005). To the best of our knowledge, no work has been done on using community mobility traces to identify social groups and places that have importance for a group of people. While place identification algorithms typically deem a place significant based on repeated patterns of user's presence at the place, identifying group members and group-place associations is much harder because informal groups do not have a clear pattern in terms of group meeting times, group composition or group member attendance. Therefore, GPI relies on repeated user copresence at the same place to determine the group members, and consequently the meeting places. The underlying assumption is that group members have a much higher Degree of Copresence (DCP) than non-group members (i.e. the DCP is defined as the total number of times two members were copresent divided by the total number of group meetings). The fact that group members are typically present only to a fraction of the meetings and non-group members can possibly be present at meetings raises the following question: What is the required DCP between group members considered by GPI?

We performed a theoretical analysis that determined the optimal required DCP that allows GPI to balance the trade-off between group member identification percentage and false positives percentage (i.e. non-group members wrongly identified as group members). Based on this analysis, we also calculated the expected results of the GPI algorithm. We also implemented GPI and ran extensive simulations. The results were in tune with the expected theoretical values. GPI was able to identify between 90 and 96% of group members with negligible false positives when the average meeting attendance was at least 50%.

Finally, we used the GPI implementation to identify groups and places on our campus using mobility traces collected from students and faculty. To successfully integrate GPI into a mobile computing and communication infrastructure, it is essential that this infrastructure provides support to collect accurate and continuous user location data both indoors and outdoors. The hardware infrastructure has to be cheap and easily deployable in order to enable location collection across large areas; as such, software solutions that take advantage of existing hardware infrastructure are preferable. Furthermore, systems that compute the location on mobile devices and allow users to decide when and what parts of the mobility traces are shared encourage the early technology adoption for privacy-conscious users.

Considering these requirements, we chose the WiFi-based Intel PlaceLab (LaMarca et al., 2005) location engine that computes location on mobile devices using the position and signal strength of visible access points. This system takes

advantage of existing access points, which are relatively densely deployed in cities. Therefore, it can work both indoors and outdoors across large urban areas. In our campus, we have at least three visible access points almost everywhere, and consequently, we obtained an accuracy of 10–15 m, which is good enough for GPI. However, one major concern with this location engine is that it could cause significant battery consumption, especially when location is computed and delivered to a server frequently. Our experiments (Anand et al., 2007) using iMate KJam smart phones showed that the battery lasts for about 5–6 hr when location is computed and delivered every 30 sec, which is sufficient for GPI. This result demonstrated that GPI and geo-social mobile recommendation applications are feasible with current technologies. We then collected mobility traces over a one-month period from smart phones carried by users on our campus. GPI successfully identified all groups that met regularly during that period. Additionally, the group places extracted from these traces were identified with good accuracy.

The rest of this paper is organised as follows. Section 2 presents a number of applications that motivate the importance of identifying group membership and group-place associations. Section 3 describes our algorithm. Section 4 presents the theoretical analysis and provides guidelines for setting the constants of our algorithm function of the environment conditions. Section 5 shows simulation and experimental results. Related work is discussed in Section 6, and this paper concludes in Section 7.

2 Motivation

This section considers a college campus scenario to illustrate the two main categories of applications that can benefit from information about informal social group membership and place-group associations. The GPI algorithm can assist recommendation applications with information about groups, such as

1 members and their profile information

2 type, which can possibly be inferred from user profiles and the meeting place

3 meeting times.

Additionally, it can provide information about places, such as

1 types of groups meeting at a place and their corresponding meeting times

2 statistical information about groups that meet at a place, such as the total number of groups and the average size of groups.

Group/person recommendations

- *For students*: group membership information is leveraged to build social networks. For example, if a student needs help with a math assignment, an application can analyse her social network and discover that one of the members of her poetry reading group has a friend who is a math major; subsequently, the math major will be recommended to the person who needs help. A different application is social matching that provides recommendations for dating partners on campus. For instance, people who are members of the same groups are excluded from recommendations (i.e. they know each other already), while people who are members in similar groups and visit similar places are higher ranked in recommendations.
- *For faculty*: a faculty member looking to recruit new students to work in his/her lab is recommended a group of students who meet routinely to discuss research papers.
- *For administration*: the identified groups are used for group-centric information dissemination. For example, research groups are notified about upcoming seminars in their research area, and groups of students regularly present on basketball courts are notified about an upcoming intra-mural basketball tournament.

Place recommendations

- *For students*: a new student finds out information about popular spots for social activities on campus. For instance, a CS student could find out that the game room of the student centre is generally occupied by other CS students on Tuesday evenings.
- *For faculty*: a faculty member uses information about the places where students from his department hang out to post fliers about an upcoming course.
- *For administration*: the administration discovers places that need improvement on campus by checking the statistical information about places (e.g. type, size and demographics of the groups that meet at a place). For instance, the settings and ambiance in certain rooms of the student centre can be modified according to the number of students who spend time there.

3 The GPI algorithm

GPI takes as input the users' mobility traces obtained via any location technology. The mobility traces of the users consist of an array of location points indexed by time. To have enough data for GPI, mobility traces should be collected over an extended period of time. The goal of GPI is to analyse these traces to identify the members of informal groups and the meeting places of these groups. To understand what type of group information GPI can extract from mobility traces, we start by presenting a characterisation of typical informal groups.

- *Member structure*: the number of group members can vary greatly. For instance, a study group could have 3–5 members, a basketball group could have 10–15 people, and a group of people attending routinely seminars on wireless networks could go up to 30–50 people. Additionally, members are typically shared among groups, and they join and leave groups frequently.

- *Member attendance*: group members do not have a pattern for meeting attendance, with the attendance frequency typically varying form 100% to 50%. Consequently, the number of members at the group meetings keeps varying over time.
- *Meeting time*: unlike with the formal groups, there is no guarantee that informal groups meet regularly (e.g. weekly at the same time).
- *Meeting place*: groups are expected to share meeting places over time, such as different study groups in the library. Even worse, different groups can meet at the same place simultaneously. For instance, two different groups of students regularly have lunch in the same part of the cafeteria.

Since these characteristics emphasise the lack of patterns of informal groups, we decided that the only characteristic amenable to automatic identification is member copresence at the group place. Routine copresence among group members is almost guaranteed even though it might vary over time. Therefore, GPI's challenge is to first detect repeated copresence among users and then to analyse it to determine the group members and the group places.

Figure 1 presents the pseudo-code for our algorithm. GPI starts by identifying the important places for individual users. For this purpose, we use the clustering algorithm proposed by Kang et al. (2004). This algorithm performs time-based clustering on users' mobility traces; it starts by analysing the trace points ordered by timestamps and adds them to a cluster as long as the next point is within a permissible distance d of the existing cluster. The cluster is closed if the trace points move away from it. If the duration of such a cluster is significant (more than time t), the cluster represents a significant visit. The newly identified place is represented by the average of the geographical coordinates of these points. We set the distance threshold d to 30 m and the time threshold t to 10 min as recommended by the authors.

For each place that a user (say u_i) visited, we check if there are groups associated with this place. The function

Figure 1 GPI algorithm pseudo-code

Inputs
$U = (u_1 \ldots u_n) \rightarrow$ Input set of all users
$M = (m_1 \ldots m_n) \rightarrow$ Mobility traces for users $(u_1 \ldots u_n)$

Constants
$t \rightarrow$ Minimum time duration for significant cluster
$d \rightarrow$ Maximum distance between clusters
$d_{cp} \rightarrow$ Maximum distance between copresent users
$t_{cp} \rightarrow$ Minimum time overlap for user visits to determine copresence
$MI \rightarrow$ Maximum number of iterations
$EVF \rightarrow$ Estimated group member visit frequency
$RCP \rightarrow$ Required degree of copresence to determine a group member
$MVC \rightarrow$ Minimum visit count to determine a potential group place

The Algorithm

For each user u_i in U
  SP_i = IndividualPlaces(m_i, t, d) /* *Set of significant places for* u_i */
  For each place P_{ij} in SP_i
    DGM = empty /* *Set of discovered group members* */
    $CI = 1$ /* *Current number of iterations to identify group members* */
    Call IdentifyGroupMembers (u_i, P_{ij})
    While $CI \leq MI$
      Pick u_k /* *Random unprocessed user from* DGM */
      Call IdentifyGroupMembers (u_k, P_{ij})
      $CI = CI + 1$
    Call IdentifyMultipleGroups(DGM)
    Output DGM

Function IdentifyGroupMembers(u_i, P)
  NV = NumberOfVisits(u_i, P)
  If $NV \geq MVC$
    $EGM = NV/EVF$ /* *Estimated total group meetings* */
    Add u_i to DGM
    For each user u_k in U
      CP = CoPresenceCount($u_i, u_k, P, d_{cp}, t_{cp}$)
      If $RCP \leq CP/EGM$
        Add u_k to DGM
  Remove data for user u_i at place P from SP_i
  Mark u_i as processed in DGM

IdentifyGroupMembers uses copresence information to identify the group members. This function first checks if the user u_i has a significant number of visits at the place (say P) to ensure that the algorithm has sufficient visit data for analysis. This is done by setting a constant for the minimum number of visits, Minimum Visit Count (MVC). Setting constants in GPI is an essential part of the algorithm given the volatile nature of informal social groups. With changing operational environments, the constants can be set differently to achieve better performance. Section 4 discusses the criteria used to set the values of all constants in GPI. If the number of visits of u_i at P is at least MVC, the function calculates the estimated number of group meetings based on the Estimated Visit Frequency (EVF). Estimation of the group meetings is required because it is not possible to determine the actual number of group meetings from the place visit data of a user.

Next, for each other user u_k, the function analyses her place visit data to check potential copresence with u_i at P. This information is used to build a copresence matrix with respect to u_i and P as illustrated in Table 1. For copresence to be considered in the matrix, the distance between the identified places for two users should be less than d_{cp} and the time overlap between the visits should be at least t_{cp}. The function uses the copresence matrix to compute the DCP of u_i with all the other users. The DCP is defined as the total number of times two users are copresent divided by the total number of group meetings. If the calculated DCP between u_i and u_k is greater than the Required Degree of Copresence (RCP), u_k is added to the set Discovered Group Members (DGM). Finally, the function removes the data for u_i at place P and marks the user as processed such that the algorithm will not analyse u_i at P again.

Table 1 Copresence matrix for user u_i at place P, wherein 1 implies copresence with another user and 0 otherwise

Visit number (u_i at P)	u_1	u_2	u_3	u_4
1	1	1	1	1
2	0	1	1	0
3	1	1	0	0
4	1	1	0	1
5	1	0	1	0
6	0	1	0	1

In the main part of the algorithm, the function *IdentifyGroupMembers* is repeated with an unprocessed user from the set DGM to discover more group members. This is necessary because it is possible that certain members were not present at the group meetings when the first user was present, but they were sufficiently copresent with the new user picked up in this iteration. However, the probability of encountering such users decreases significantly with every subsequent iteration. To speed up the running time, this process is repeated for Maximum Iterations (MI) times (less than the number of users) because no new group members are expected to be identified if more iterations are executed.

Finally, GPI analyses DGM to check for multiple groups at the same place, by calling *IdentifyMultipleGroups*. In rare cases, it is possible that the users in DGM belong to two or more different groups at the same place. This may happen when there are multiple groups at the same place, and several shared members have sufficient copresence with members of all the groups. However, it is easy to detect and divide such groups considering the observation that besides the shared members, members of one group do not have enough copresence with members of another group. For example, suppose that there are two groups (u_1, u_2, u_3) and (u_3, u_4, u_5) that routinely hang out at the same place P. Then u_1 and u_2 have significant copresence with each other and u_3, but not with u_4 and u_5. Similarly u_4 and u_5 have significant copresence only with each other and u_3. We successfully tested our procedure to split groups, but we do not present the details due to the lack of space.

Once the algorithm completes, we need to define the identified group place P. We compute the average of the geographical coordinates of all trace points of all visits by all users at P (let this be C). C is defined as a point, but most applications are interested in well-defined places rather than points. P is determined by looking at the actual geographies around the point C. For example, if C falls inside an office building, P is defined as all the rooms that overlap with a circular area of radius E around C, where E is the maximum error in determining C (i.e. this error is introduced by the location technology). If the application needs to associate a place with only one room, then P is considered to be the room that contains C.

GPI executes off-line, and as such, its running time is not essential for the applications. Nevertheless, we analysed its complexity to estimate how long it would take to identify groups and places for a large user population. The asymptotic running time of the algorithm is $O(n^2 \times v^2 + nt)$, where n is the number of users, v is the maximum number of significant visits for a user and t is the maximum number of mobility trace points for a user. For instance, let us assume that the user population is 10, 000, and we collect location data for every user at every 10 sec, for 6 hr a day, during one month period. Running on a medium size server, GPI will complete in several hours, which is acceptable considering that it is executed rarely.

4 Analysis of constants in GPI

As discussed in the previous section, GPI uses six constants (RCP, MVC, EVF, MI, d_{cp}, t_{cp}) that affect significantly the performance of the algorithm. It is important to note that the values of these constants do not change once they have been set for a certain environment. However, with changing operational environments, it is possible to achieve better identification results by altering the values of these constants. For example, if we know that people meet more frequently in a particular environment, we can set the estimated group member visit frequency, EVF, higher. Similarly, we can set the MVC higher, if we know that groups meet very frequently.

Our goal in this section is to provide the reader with an understanding of how these constants affect the algorithm,

a theoretical analysis that can be used to alter the values of these constants with changing environments and guidelines for setting these values such that the algorithm works well in most situations. We start with several definitions and lemmas that will be used in our analysis. For the sake of brevity, we omit the straightforward proofs of the lemmas.

Definition 1: *The expected number of visits at a group meeting by a user X is defined as $XF \times TGM$, where XF is the visit frequency and TGM is the total number of group meetings.*

Definition 2: *We define two random variables, CP and Degree of Copresence (DCP), as follows:*

- *$CP = 1$ if users X and Y are copresent at a group meeting and 0 otherwise.*
- *DCP, the DCP between X and Y w.r.t. TGM, is defined as the number of times X and Y were copresent divided by the total number of group meetings.*

Lemma 1: *At any group meeting, the probability that X and Y are copresent is $P[CP = 1] = XF \times YF$, and the probability that they are not copresent is $P[CP = 0] = 1 - XF \times YF$.*

Lemma 2: *The expected DCP between X and Y w.r.t. the total number of group meetings is $E[DCP] = XF \times YF$.*

Lemma 3: *The probability that the DCP between X and Y w.r.t. TGM is at most Δ is given by $P[DCP \leq \Delta] = \sum_{i=0}^{\Delta \times TGM} \binom{TGM}{i} \times (P[CP = 1])^i \times (P[CP = 0])^{TGM-i}$.*

Lemma 4: *The probability that the DCP between X and Y with respect to TGM is at least Δ is given by $P[DCP \geq \Delta] = \sum_{i=\Delta \times TGM}^{TGM} \binom{TGM}{i} \times (P[CP = 1])^i \times (P[CP = 0])^{TGM-i}$.*

4.1 Required degree of copresence (RCP)

GPI assumes that group members must have a DCP of at least RCP. Finding an ideal value for RCP is hard as the DCP among group members and between a group member and a nongroup member varies with different groups. Using the assumption that group members would generally be copresent more than non-group members, the ideal value of RCP should be set such that the RCP for all group members is greater than RCP and for non-group members is less than RCP. As Lemmas 3 and 4 show, this degree is a function of the frequency of group place visits; generally, this frequency is higher for a group member (GMF) than a non-group member (NGMF).

Let us assume that all non-group members have NGMF= 0.1 and all group members have GMF = 0.5. These values are relatively high for non-group members and low for group members. As such, this example is close to a worst case scenario. Using Lemma 3, given the total number of group meetings and the visit frequencies of two members, we can calculate the probability that DCP is greater than a certain value Δ. For example, let us consider a random group member X and TGM= 20. The probability that the DCP between X and any other group member is greater than 0.2 is 0.77. Therefore, since X was selected randomly, we expect to identify 77% of the group members. Note that this analysis also shows that the identification percentage is independent of the group size. In the same way, we compute the probability that the DCP between a non-group member and any group member is greater than RCP= 0.2. This probability gives us the percentage of false positives (i.e. non-group members wrongly identified as group members), which in this case is only 1.5%.

As we will discuss later in this section and in the following section, the identification percentage is much higher (up to 98%) when GMF is between 0.7 and 0.9. We can achieve similar results even for GMF= 0.5, as shown in Figure 2 by setting RCP= 0.1, but the number of false positives increases significantly in this case (Figure 3). A better solution for a higher identification percentage, while maintaining a low percentage of false positives, is to run the algorithm for several iterations as explained in Section 3.

Figure 2 Expected percentage of group members identified in the first iteration

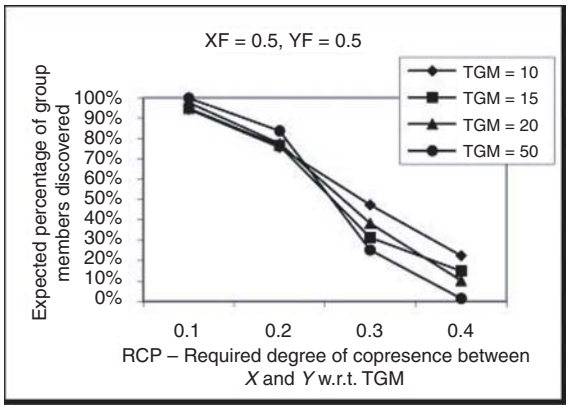
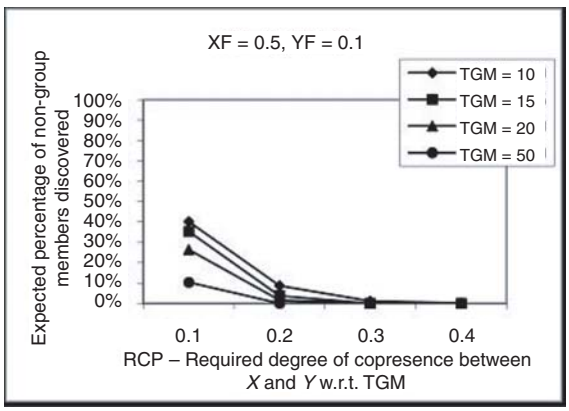


Figure 3 Expected percentage of false positives in the first iteration



For instance, the identification percentage goes up from 77% to 90% after the second iteration. Let us assume that a group member X was identified in the first iteration and another group member Y was not identified. If we start the second iteration with X, the probability that Y is identified in the

second iteration can be computed as the product of three terms:

1 the probability that X was identified in the first iteration

2 the probability that Y was not identified in the first iteration (obtained using Lemma 3)

3 the probability that Y is identified in second iteration.

For TGM= 20, GMF= 0.5 and RCP= 0.2, this product is 0.13. Therefore, we expect to pick up 13% more group members in the second iteration and about 1.2% more non-group members. Figures 4 and 5 show the results after two iterations while varying RCP and TGM. We note that as TGM increases, GPI is expected to pick up more members when the required RCP is close to the expected DCP (the expected degree can be calculated using Lemma 4). This result is explained by the law of large numbers (Casella and Berger, 2001). We performed a similar analysis for the next few iterations and discovered that the expected identification percentage of group members goes up by only 2% in the third iteration and does not increase significantly after that.

Figure 4 Expected percentage of group members identified in the second iteration

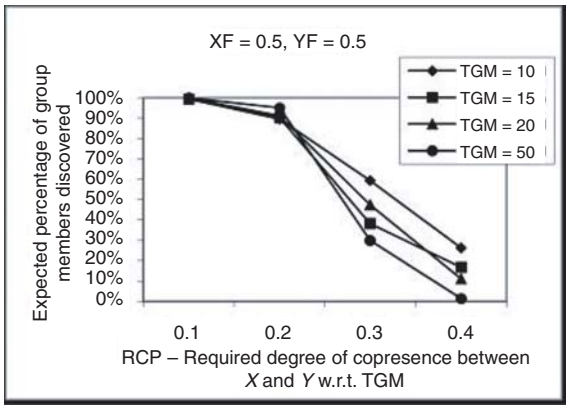
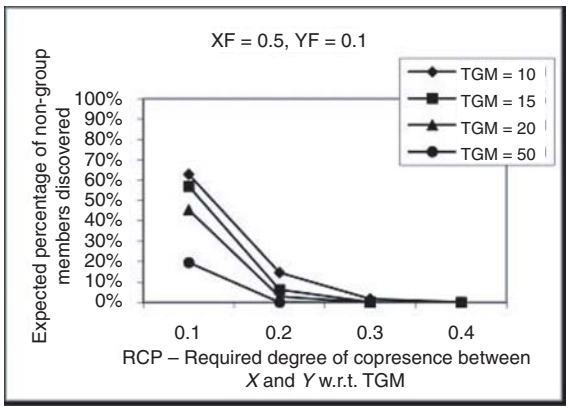


Figure 5 Expected percentage of false positives in the second iteration



Considering the results presented so far and the trade-off between high identification percentage and low false positive percentage, RCP= 0.2 provides the best results when GMF= 0.5 and NGMF= 0.1. We performed a similar analysis, by selecting different visit frequencies for group members. Table 2 shows the acceptable RCP values for each of these cases, where the criteria for acceptance are: more than 85% of group members and less than 3% of non-group members are expected to be identified in the first iteration, and more than 95% of group members and less than 6% of non-group members are expected to be identified by the third iteration. These results and the simulation results from the following section made us decide that RCP= 0.2 is a value that works well in most situations.

Table 2 Acceptable values for RCP when TGM= 20

GMF (XF, YF)	*RCP*
(0.7, 0.5)	0.2 – 0.25
(0.7, 0.7)	0.2 – 0.4
(0.9, 0.5)	0.25 – 0.35
(0.9, 0.7)	0.25 – 0.5
(0.9, 0.9)	0.25 – 0.7

4.2 Estimated group member visit frequency (EVF)

EVF is computed as the number of times a group member attends a group meeting divided by the total number of group meetings. Using this value and the actual number of user visits, we compute the estimated total number of group meetings EGM. For example if the actual number of user visits at a place is 10 and the estimated group member visit frequency is 0.5, then the estimated total number of group meetings will be 20. Note that when we analyse users' visit data, it is not possible to determine the actual visit frequency of group members and the total number of group meetings.

We recommend setting EVF to the mean value of 0.5 since this value implies the least expected variation in the performance of the algorithm. Note that EVF is used to estimate EGM, which is further used to calculate DCP and compare it with RCP. If EVF= 0.5 and the actual visit frequency is 0.7, we overestimate the total number of group meetings EGM. Therefore, if RCP= 0.2, the algorithm would behave as if RCP was set to 0.28. Similarly, if actual visit frequency is 0.9, the effective value of RCP is 0.36. Comparing these values with the results from Table 2, we observe that they are within the range of acceptable values for RCP.

4.3 Minimum visit count (MVC)

MVC denotes the minimum number of visits a user should have at a place before analysing it as a group place; it ensures sufficient user presence at the potential group place and sufficient data for analysis. As Figure 3 shows, the performance deteriorates significantly in terms of the expected percentage of identified group members when TGM= 10. This is primarily due to smaller data sets, which lead to higher variance from the expected value. The performance is, however, more stable when TGM is 15 or higher. Therefore, we recommend setting MVC to 7.

4.4 Maximum iterations (MI)

By extending the analysis presented in Figures 2–5, we observed that subsequent iterations after the third one do not significantly increase the expected identification percentage, but on the other hand, increase the expected percentage of false positives. Therefore, we recommend setting the maximum number of iterations MI to 3.

4.5 Maximum distance between copresent users (d_{cp})

d_{cp} is the maximum distance between copresent users and is affected by two factors. The first is the actual geography of the place where the algorithm operates. For example, if the group places are big halls, even users that are quite far apart are copresent, and therefore, d_{cp} should be set high. On the other hand, if the places are regular size offices, setting d_{cp} too high might wrongly suggest copresence. The second factor is the accuracy of the location engine. For example, if the location engine has an average accuracy of E metres, it is possible that two users who are actually D metres apart could be detected at a distance of $2E + D$. While D might be less than d_{cp}, $D + 2E$ may be greater than d_{cp} and copresence might not be detected. Since the geography of the place is hard to quantify, we recommend to set d_{cp} to 30 m based on the accuracy of our location engine. The same threshold is recommended by Kang et al. (2004) when using the same location engine.

4.6 Minimum time overlap for copresent visits (t_{cp})

t_{cp} is the minimum time overlap between visits at the same place by two users such that the users are considered copresent. An optimal value depends on the typical duration of an informal group meeting. Assuming that a group meeting would last at least 20 min, and users would be present for at least 50% of the time, we recommend setting t_{cp} to 10 min.

5 Evaluation

The goals of GPI are to achieve

1 high percentage of group member identification, while maintaining a low percentage of false positives
2 identify the place of the group meetings with good accuracy.

To demonstrate the performance of GPI with respect to these two goals, we first ran simulations to get group identification results for a large population of users under various scenarios. These results matched well with the expected values derived from the theoretical analysis presented in the previous section. Then, we obtained experimental results by running the algorithm over mobility traces collected using smart phones carried by users on campus for one month. These experiments validated the theoretical and simulation results for group identification. They also showed that the average place identification error was less than the error introduced by the location technology.

5.1 Simulations

5.1.1 Setup

We use 40 users ($U_1 \ldots U_{40}$), 5 groups ($G_1 \ldots G_5$) and 10 places ($P_1 \ldots P_{10}$) with minimum distance of 30 m between each place. At any point, a user can be in any of the ten possible places, according to their mobility traces. These traces are generated randomly, while taking into account GMF and NGMF. If for instance GMF= 0.5 and NGMF= 0.1, a group member is present at the group place with probability 0.5 (we assume that all meetings for a group take place at the same place) and at any of the remaining nine places with probability 0.5. Similarly, a non-group member is present at the group place with probability 0.1 and at any of the remaining nine places with probability 0.9. The groups are classified into three categories:

1 groups that do not share users and meeting places with other groups (G_1)
2 groups that share users with other groups (G_2 and G_3)
3 groups that share users as well as meeting places with other groups (G_4 and G_5).

The composition of the groups is as follows.

- G_1 has members ($U_1 ... U_{10}$) and meeting place P_1.
- G_2 has members ($U_{11} ... U_{20}$) and meeting place P_2.
- G_3 has members ($U_{16} ... U_{25}$) and meeting place P_3.
- G_4 has members($U_{26} ... U_{35}$) and meeting place P_4.
- G_5 has members($U_{31} ... U_{40}$) and meeting place P_4.

Note that we do not vary the size of the groups because the theoretical analysis demonstrated that the group size does not affect the identification or false positives percentages. The meeting time for each group is generated randomly, while ensuring that two groups that share members do not meet at the same time.

5.1.2 Results

We set the GPI constants according to the recommendations from the previous section (EVF= 0.5, MVC= 7, $d_{\text{cp}} = 30$, $t_{\text{cp}} = 10$, MI= 3). Figures 6 and 7 show the identification percentage and the false positive percentage, respectively, as function of RCP for TGM= 20. Figures 8 and 9 present results for TGM= 50. The plotted curves are for GMF set to 0.3, 0.5, 0.7 and 0.9. We set NGMF to 0.1 in all cases. From these graphs, we observe that:

- Setting RCP= 0.2, as recommended in the previous section, yields the best overall performance in terms of identification percentage and false positives percentage.
- When GMF is over 0.7, GPI identifies almost all group members (over 96%), while the false positives are under 1%. Even in the very unlikely case when GMF= 0.3, GPI still indentifies over 70% of the members, with 7% false positives.

- GPI performs better when the groups meet more times. As TGM increases, the identification percentage increases, and the false positives percentage decreases.
- These results are in tune with the theoretical analysis. The maximum variation is $6 - 8\%$, which is expected given the relatively small size of our user population.

Figure 6 Average percentage of group members identified when TGM = 20

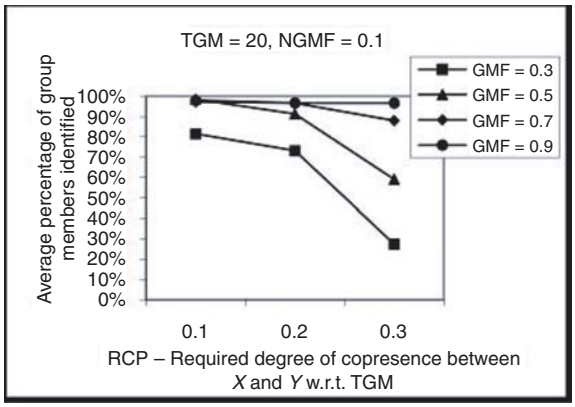


Figure 7 Average percentage of false positives when TGM = 20

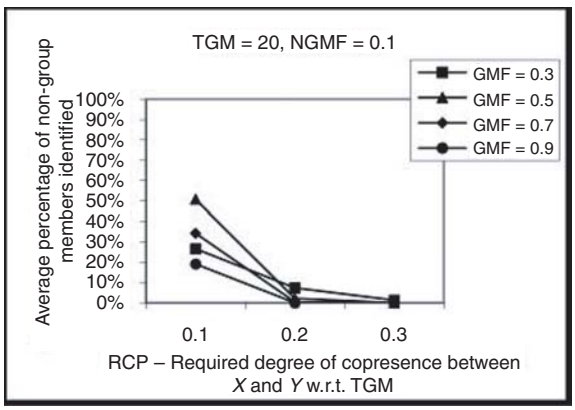


Figure 8 Average percentage of group members identified when TGM = 50

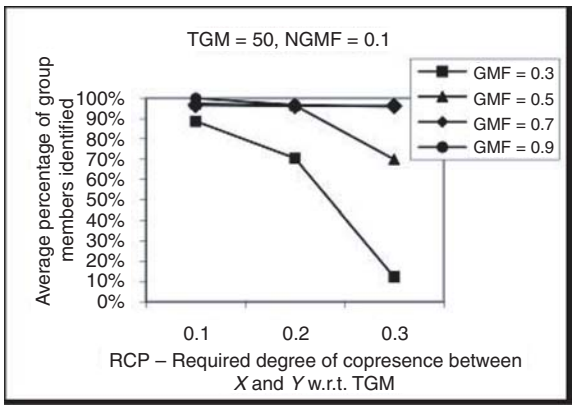
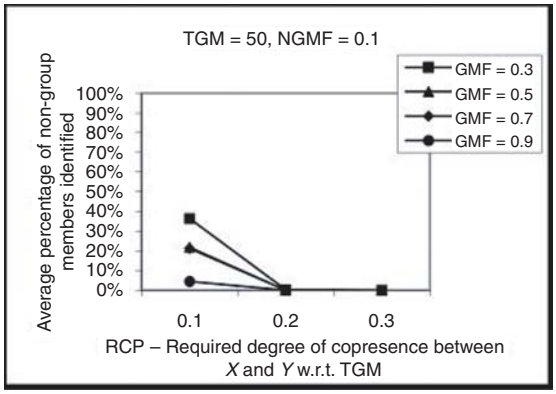


Figure 9 Average percentage of false positives when TGM = 50



5.2 *Experimental evaluation*

5.2.1 *Setup*

To collect location data, we used HTC TyTN smart phones running the WiFi-based Intel's PlaceLab (LaMarca et al., 2005) location engine. The mobility traces were collected on our university campus, which typically has at least three visible WiFi access points at every place. The location accuracy ranged from 10 to 15 m. The phone battery lasted for about 5–6 hr with the location computation frequency set to 10 sec (note that we send this location to a server in real-time).

Seven users carried the phones for one month. For every place where they spent more than 20 min, they recorded the date and time of visit on a log sheet. This log sheet was used to verify the place visit data extracted by the algorithm. The users were part of two different groups and their composition was as follows:

- U_2, U_3 and U_4 are graduate students who are part of the same research group and routinely visit the same research lab (Lab_1). They form Group_1. U_1 is a professor who occasionally visits (Lab_1) and therefore is a non-group member. Table 3 presents the user visit data at Lab_1.
- U_5, U_6 and U_7 are graduate students who routinely visit another research lab (Lab_2) that is about 50 m away from (Lab_1). These users form Group_2.

To increase the user population, one of the authors carried multiple phones representing 'dummy' users for about a month. For each user, we had a preset visit frequency at the group place. For each day, the visits were determined by a random visit generator similar to the one used in the simulations. There were 16 users $U_8 \ldots U_{23}$ with the following visit patterns.

- $U_8 \ldots U_{11}$ form Group_3. They have GMF= 0.5 for U_8 and U_9 and GMF= 0.7 for U_{10} and U_{11}. U_{13} and U_{14} are non-group members with NGMF= 0.1. U_{15} is a non-group member with NGMF= 0.2.
- $U_{16} \ldots U_{20}$ form Group_4 and have GMF= 0.6. $U_{21} \ldots U_{23}$ are non-group members with NGMF = 0.1.

Table 3 Visit data at Lab_1

Date	U_1	U_2	U_3	U_4
28-June	2 PM – 5 PM	3 PM – 5 PM	2 PM – 5 PM	
29-June			11 AM – 5 PM	
2-July	6 PM – 9 PM	3 PM – 5 PM	10 AM – 4 PM	3 PM – 8 PM
3-July	3 PM – 8 PM	10 AM – 4 PM		5 PM – 6 PM
5-July	2 PM – 8 PM	11 AM – 5 PM		3 PM – 5 PM
6-July			11 AM – 6 PM	
9-July			11 AM – 5 PM	
10-July		12 PM – 4 PM	1 PM – 5 PM	3 PM – 10 PM
11-July		9 AM – 3 PM		6 PM – 7 PM
12-July				2 PM – 4 PM
13-July			1 PM – 7 PM	3 PM – 6 PM
17-July		11 AM — 3 PM		
18-July		11 AM – 5 PM	3 PM – 5 PM	
19-July		10 AM – 5 PM	4 PM – 10 PM	
23-July				3 PM – 5 PM
24-July		10 AM – 1 PM		3 PM – 10 PM
25-July		11 AM – 3 PM	11 AM – 5 PM	2 PM – 5 PM
27-July		9 AM – 3 PM		2 PM – 5 PM

5.2.2 *Results*

We tested the algorithm by setting RCP to 0.1 and 0.2 to see if the results match the theoretical analysis and the simulation study. The other three constants d_{cp}, EVF and MVC were set to 30, 0.5 and 10, respectively. For RCP = 0.1, GPI identified all group members, but there were several false positives, U_1 for $Group_1$, U_{12} and U_{15} for $Group_3$ and U_{21} and U_{23} for $Group_4$. However, for RCP= 0.2, GPI worked very well, identifying all group members without any false positives.

5.3 *Accuracy of place identification*

Assuming that the accuracy of the location technology is E metres, there are two types of visits at a place:

1 The user stays in the same spot for the visit. The location points for this user will be spread in a circular area with radius E. The error between the detected location and the actual location is at most E (the worst case happens when all location points are at the same point at a distance E from the actual point).

2 The user moves in different spots ($S_1 \ldots S_n$) during the visit. The average of the geographical coordinates of these different spots is the actual visit place S. The detected location D is the average of the geographical coordinates of all the identified spots. In the worst case, all the detected points could suffer a geometric translation along a certain vector at a distance E. Therefore, the maximum distance between the detected location D and actual location S is E.

The group place is calculated as the average of the geographical coordinates of the different spots for all visits by all detected group members. The situation is similar to a single user visiting different spots, and the error can be at most E. We compared our experimental results with this bound and found a mean error of 8.6 m. This value is indeed less than the maximum expected error (10–15 m for our location engine).

6 Related work

Our system leverages work and has similarities with a number of projects in the following areas: location technologies, individual place identification and context-aware recommendation systems. Additionally, privacy (and especially location privacy) is an issue that has to be discussed due to the potentially sensitive nature of group and place recommendations.

The current suite of location technologies offer varying degrees of accuracy, availability, ease of deployment and privacy. Since users want to have control over their location for privacy reasons, we discuss just the systems that compute the location on user devices. In this way, users can decide what parts of the mobility traces are made available to an application that uses our algorithm. GPS (Enge and Misra, 1999) offers 10–15 m accuracy, but requires additional hardware and does not work indoors. Rosum's TV-GPS (Rabinowitz and Spilker, 2002) is another outdoor technology that uses digital TV synchronisation signals and provides 5–25 m accuracy. RADAR (Bahl and Padmanabhan, 2000) and (Haeberlen et al., 2004; Hightower and Borriello, 2004; Krumm and Horvitz, 2004; Ladd et al., 2002) are WiFi-based technologies that work indoors and provide 3–20 m accuracy, but require collection of significant statistical data about the operational environment. MIT's Cricket (Priyantha et al., 2000) works in indoor environments and offers accuracy of a few centimetres, but it requires additional hardware and considerable effort to deploy on a larger scale. Finally, PlaceLab (LaMarca et al., 2005) is a WiFi-based technology that provides 10–15 m accuracy when enough access points are visible form the mobile device, works both indoors and outdoors, and does not require additional hardware. PlaceLab also works using GSM signals. In our experiments we used this system

because we considered that it answers well the requirements for availability, ease of deployment and privacy, while the accuracy is reasonably good.

Some of the initial work on place extraction was done using GPS, where loss of GPS signal was used to infer important indoor locations. Marmasse and Schmandt (2000) introduced this technique, but it struggled with recognising places larger than a typical home. Ashbrook and Starner (2003) improved the mechanism to overcome this problem, but their algorithm was still unable to infer important outdoor locations or multiple places within a single building. Laasonen et al. (2004) and Hightower et al. (2005) presented fingerprinting based techniques that collect data such a GPS signals and radio beacons from WiFi towers for all the places the user visits, and then use statistical inference techniques to recognise repeated visits to the same places. Finally, Hariharan and Toyama (2004) and Kang et al. (2004) use time and location information to find places where the user spends a significant amount of time (called stays). Then, they cluster the stays and find places where a user has experienced more than one stay to infer a significant place. This technique works for us since it provides geographical location information about the significant place, time information for all the stays at the place and it works both indoors and outdoors. Both these algorithms provide similar results, but we chose the algorithm by Kang et al. (2004) because they used PlaceLab like us and their algorithm is less computationally expensive.

Context-aware recommendation systems provide information tailored to user's context (e.g. location, preferences, copresent users) or environmental context (e.g. time). GPI can enhance these systems with additional contextual information, namely user presence in a group or place. GeoNotes (Espinoza et al., 2001) allows users to post virtual notes at places, which can be read by other users visiting the same place. For instance, GPI can enhance it such that group members can post information that only other group members can read. Similarly, the location-based reminding service presented in Sohn et al. (2005) can use GPI to deliver reminders when the user is at a group place or to all members of a particular group. Heijden et al. (2005) and Yang et al. (2008) present context-aware recommender systems that can assist users with shopping. GPI can enhance them to offer information such as group discounts when copresence of a group member is detected.

Increasingly, location-aware recommendation systems such as the ones discussed above require the user to share location data, presenting a difficult privacy trade-off where disclosing location could be risky but at the same time valuable. Results from a study by Consolvo et al. (2005) show that users were willing to share information that could be useful to a requester that is socially connected to the user, depending on who was requesting it and why they were requesting it. Another study, done in Manhattan by Grandhi et al. (2005), shows that over 84% of the 500+ respondents were willing to share location with a system to obtain services such as information about occupancy and crowding in public places and over 77% were willing to share location with others in exchange of a service. However, we still need to understand user's privacy concerns and design systems to address them. A study by Marmasse and Schmandt (2000) shows that user's privacy preferences depend on the person requesting it rather than the situation in which it was requested. Barkhuus and Dey (2003) present a study that advocates the development of position-aware services that rely on the device's knowledge of its own location rather than location-tracking services that are based on other parties tracking the user's location (we used this approach). Langheinrich (2002) presents an architecture that allows users to keep track of privacy sensitive information that is used by the system. Confab, proposed in Hong and Landay (2004), is a generic toolkit that can be used to facilitate the development of privacy sensitive ubiquitous computing applications, and Myles et al. (2003) presents a system that allows users to define rules for sharing location information and thereby minimises the system-user interaction for information sharing. Krumm (2007) presents an evaluation of different mechanisms for privacy protection of location data.

7 Conclusions and future work

This paper presented GPI, an algorithm for automatic identification of informal social groups and their associated places. GPI is enabled by fundamental properties of mobile computing such as mobility and location-awareness, and its results can be used in a large spectrum of applications that provide geo-social recommendations about people, groups and places. We presented a theoretical analysis of the performance of the algorithm in terms of identification accuracy of group members and group places. The simulation results matched very closely the expected theoretical values and demonstrated that $90 - 96\%$ of group members can be identified with negligible false positives when the user meeting attendance is at least 50%.

We also demonstrated that GPI works in real-life conditions with existing technologies. In our case, we took advantage of complete WiFi coverage across the campus to compute and collect location data from smart phones distributed to students and faculty. The experimental results demonstrated that we can achieve good location accuracy, 10–15 m, and the phone battery lasts 5–6 hr when collecting location data every 30 sec. Under these conditions, GPI identified all groups that met during the one-month period of collected mobility traces. Furthermore, the place identification error was less than the error introduced by our location technology. In the near future, we will integrate GPI into our mobile social computing middleware developed for the SmartCampus project (SmartCampus, 2005). This middleware allows rapid development of mobile social applications for large user communities. We plan to have several hundred users carrying smart phones that run such applications. In this way, we will be able to perform more detailed user studies to validate GPI's performance in conjunction with geo-social recommendation applications.

Acknowledgement

This material is based upon work supported by the National Science Foundation under Grants No. CNS-0454081, IIS-0534520 and CNS-0520033. Any opinions, findings and

conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

Anand, A., Manikopoulos, C., Jones, Q. and Borcea, C. (2007) 'A quantitative analysis of power consumption for location-aware applications on smart phones', *Proceedings of the 2007 IEEE International Symposium on Industrial Electronics (2007)*, Vigo, Spain, June, pp.1986–1991.

Ashbrook, D. and Starner, T. (2003) 'Using GPS to learn significant locations and predict movement across multiple Users', *Personal and Ubiquitous Computing (2003)*, Vol. 7, No. 5, pp.275–286.

Bahl, P. and Padmanabhan, V.N. (2000) 'RADAR: an in-building RF-based user location and tracking system', *Proceedings of IEEE Infocom (2000)*, Tel Aviv, Israel, March, Vol. 2, pp.775–784.

Barkhuus, L. and Dey, A. (2003) 'Location-based services for mobile telephony: a study of user's privacy concerns', *Proceedings of INTERACT (2003)*, Zurich, Switzerland, September, pp.709–712.

Casella, G. and Berger, R. (2001) *Statistical Inference*, Duxbury Press, Pacific Grove, USA, 2001.

Consolvo, S., Smith, I., Matthews, T., Lamarca, A., Tabert, J. and Powledge, P. (2005) 'Location disclosure to social relations: why, when, and what people want to share', *Proceedings of CHI (2005)*, Portland, USA, April, pp.81–90.

Enge, P. and Misra, P.(1999) 'The global positioning system', *IEEE (Special Issue on GPS)*, Vol. 87, No. 1, pp.3–172.

Espinoza, F., Persson, P., Sandin, A., Nystr, H., Cacciatore, E. and Bylund, M. (2001) 'GeoNotes: social and navigational aspects of location-based information systems', *Proceedings of International Conference on Ubiquitous Computing (2001)*, Atlanta, USA, October, pp.2–17.

'Facebook' (2004) Available at: http://www.facebook.com.

Grandhi, S.A., Jones, Q. and Karam, S. (2005) 'Sharing the BigApple: a survey study of people, place, and locatability', *Proceedings of CHI (2005)*, Portland, USA, April, pp.1407–1410.

Haeberlen, A., Flannery, E., Ladd, A.M., Rudys, A., Wallach, D.S. and Kavraki, L.E. (2004) 'Practical robust localization over large-scale 802.11 wireless networks', *Proceedings of ACM MobiCom (2004)*, Philadelphia, PA, September, pp.70–84.

Hariharan, R. and Toyama, K. (2004) 'Project lachesis: parsing and modeling location histories', *Proceedings of 3rd International Conference on Geographic Information Science (2004)*, Adelphi, USA, October, pp.106–124.

Heijden, H., Kotsis, G. and Kronsteiner, R. (2005) 'Mobile recommendation systems for decision making on the go', *Proceedings of International Conference on Mobile Business (2005)*, Sydney, Australia, July, pp.137–143.

Hightower, J. and Borriello, G. (2004) 'Accurate, flexible, and practical location estimation for ubiquitous computing', *Proceedings of International Conference on Ubiquitous Computing (2004)*, Nottingham, England, September, pp.88–106.

Hightower, J., Consolvo, S., LaMarca, A., Smith, I., Hughes, J. and Borriello, G. (2005) 'Learning and recognizing the places we go', *Proceedings of International Conference on Ubiquitous Computing (2005)*, Tokyo, Japan, September, pp.159–176.

Hong, J. and Landay, J. (2004) 'An architecture for privacy-sensitive ubiquitous computing', *Proceedings of the 2nd International Conference on Mobile Systems, Applications, and Services (2004)*, Boston, USA, June, pp.177–189.

Jones, Q., Grandhi, S.A., Terveen, L. and Whittaker, S. (2004) 'People-to-people-to-geographical-places: the P3 framework for location-based community systems', *Computer Supported Cooperative Work (2004)*, Chicago, USA, November, pp.249–282.

Jones, Q., Grandhi, S.A., Karam, S., Whittaker, S., Zhou, C. and Terveen, L. (2007) 'Geographic place and community information preferences', *Computer Supported Cooperative Work (2007)*, July.

Kang, J.H., Welbourne, W., Stewart, B. and Borriello, G. (2004) 'Extracting places from traces of location', *Proceedings of the International Workshop on Wireless Mobile Applications and Services on WLAN (2004)*, Philadelphia, USA, October, pp.110–118.

Krumm, J. (2007) 'Inference attacks on location tracks', *Proceedings of Fifth International Conference on Pervasive Computing (2007)*, Toronto, Canada, May, pp.127–143.

Krumm, J. and Horvitz, E. (2004) 'Locadio: inferring motion and location from Wi-Fi signal strengths', *Proceedings of International Conference on Mobile and Ubiquitous Systems: Networking and Services (2004)*, Boston, USA, August, pp.4–13.

Laasonen, K., Raento, M. and Toivonen, H. (2004) 'Adaptive on-device location recognition', *Proceedings of the Second International Conference on Pervasive Computing (2004)*, Vienna, Austria, April, pp.287–304.

Ladd, A.M., Bekris, K.E., Rudys, A., Marceau, G. and Kavraki, L.E. (2002) 'Robotics-based location sensing using wireless ethernet', *Proceedings of ACM MobiCom (2002)*, Atlanta, USA, September, pp.227–238.

LaMarca, A., Chawathe, Y., Consolvo, S., Hightower, J., Smith, I., Scott, J., Sohn, T., Howard, J., Hughes, J., Potter, F., Tabert, J., Powledge, P., Borriello, G. and Schilit, B. (2005) 'PlaceLab: device positioning using radio beacons in the wild', *Proceedings of the Third International Conference on Pervasive Computing (2005)*, Munich, Germany, May, pp.116–133.

Langheinrich, M. (2002) 'A privacy awareness system for ubiquitous computing environments', *Proceedings of the 4th International Conference on Ubiquitous Computing (2002)*, Goteborg, Sweden, October, pp.315–320.

'LinkedIn' (2002) Available at: http://www.linkedin.com/.

Marmasse, N. and Schmandt, C. (2000) 'Location-aware information delivery with comMotion', *Proceedings of the Second International Symposium on Handheld and Ubiquitous Computing (2000)*, Bristol, UK, September, pp.157–171.

Myles, G., Friday, A. and Davies, N. (2003) 'Preserving privacy in environments with location-based applications', *IEEE Pervasive Computing, 2003*, Vol. 2, No. 1, pp.56–64.

'MySpace' (2003) Available at: http://www.mysace.com.

Priyantha, N.B., Chakraborty, A. and Balakrishnan, H. (2000) 'The cricket location-support system', *Proceedings of ACM MobiCom (2000)*, Boston, USA, August, pp.32–43.

Rabinowitz, M. and Spilker, J. (2002) 'A new positioning system using television synchronization signals', Available at: http://www.rosum.com/.

'SmartCampus' (2005) Available at: http://smartcampus.njit.edu/.

Sohn, T., Li, K.A., Lee, G., Smith, I.E., Scott, J. and Griswold, W.G. (2005) 'Place-Its: a study of location-based reminders on mobile phones', *Proceedings of International Conference on Ubiquitous Computing (2005)*, Tokyo, Japan, September, pp.232–250.

Takeuchi, Y. and Sugimoto, M. (2005) 'An outdoor recommendation system based on user location history', *Proceedings of the 1st International Workshop on Personalized Context Modeling and Management for UbiComp Applications (2005)*, Tokyo, Japan, September, pp.91–100.

Yang, W.S., Cheng, H.C. and Dia, J.B. (2008) 'A location-aware recommender system for mobile shopping environments', *Expert Systems with Applications: An International Journal, 2008*, Vol. 34, No. 1, pp.437–445.