Pers Ubiquit Comput (2003) 7: 275–286
DOI 10.1007/s00779-003-0240-0

ORIGINAL ARTICLE

Daniel Ashbrook · Thad Starner

Using GPS to learn significant locations and predict movement across multiple users

Received: 2 February 2003 / Accepted: 2 April 2003 / Published online: 11 September 2003

Abstract Wearable computers have the potential to act as intelligent agents in everyday life and to assist the user in a variety of tasks, using context to determine how to act. Location is the most common form of context used by these agents to determine the user's task. However, another potential use of location context is the creation of a predictive model of the user's future movements. We present a system that automatically clusters GPS data taken over an extended period of time into meaningful locations at multiple scales. These locations are then incorporated into a Markov model that can be consulted for use with a variety of applications in both single-user and collaborative scenarios.

Keywords Context · GPS · Location aware computing · Schedule prediction

1 Introduction

For any user-assisting technology to be truly useful and not merely irritating, it must have some knowledge of the user to be assisted: it must understand—or at least predict—what the user will do, when and where she will do it, and, ideally, the reason for her actions. User modelling is a necessary step toward gaining this understanding.

Csinger defines user modelling as "...the acquisition or exploitation of explicit, consultable models of either the human users of systems or the computational agents which constitute the systems" [4]. This definition, however, raises the question of "what constitutes a model?" For the purposes of our research, we consider a model to be a collection of data on some particular aspect of a human user's behaviour that, when associated with a limited set of contextual clues, yields predictions on what behaviour the human will engage in next.

In this paper, we describe research investigating one facet of user modelling, that of location. Location is one of the most commonly used forms of context: it is usually easy to collect location data, and other pieces of context may be inferred from location, such as the presence of other people. In our research, we used off-the-shelf Global Positioning System (GPS) hardware to collect location data in a simple and reliable manner. We constructed software to interpret the collected data, allowing the creation and querying of location models.

1.1 Previous work

In his master's thesis [15], Jon Orwant describes Doppelgänger, a "user modelling shell" that learns to predict a user's likes and dislikes. Orwant uses active badges, Unix logins and schedule files to guess where in a building a particular user is likely to go. The possible locations in the Doppelgänger system were, in a sense, hard-coded, since a user was detected by fixed locations in the infrastructure. In contrast, GPS requires no infrastructure (or, rather, its infrastructure is worldwide) but does not work inside buildings or other places where its satellite signals are not visible. Overall, however, GPS offers a wider range of location information than do infrastructure-dependent fixed sensors.

Sparacino used infrared beacons to create individualised models of museum visitors [17] allowing each exhibit to present custom audiovisual narrations to each user. As visitors move throughout the museum, their exhibit-viewing habits are classified into one of three categories: greedy (wanting in-depth information on everything), selective (wanting in-depth information on a selection of exhibits), or busy (wanting to see a little bit of everything). These classifications are estimated by a Bayesian network, using the viewer's stopping time at each exhibit as input.

Location prediction systems have become of interest in the cellular network community in recent years. The

D. Ashbrook (✉) · T. Starner (✉)
College Of Computing,
Georgia Institute of Technology,
Atlanta, GA, 30332-0280, USA
E-mail: anjiro@cc.gatech.edu
E-mail: thad@cc.gatech.edu

United States government wants to be able to locate people who place emergency 911 calls from cell phones, and various location-based contextual services are being discussed. Another concern is limiting the amount of cellular infrastructure dedicated to locating a user so her calls may be delivered. Bhattacharya and Das describe a cellular-user tracking system for call delivery that uses transitions between wireless cells as input to a Markov model [2]. As users move between cells, or stay in a cell for a long period of time, the model is updated and the network has to try fewer cells to successfully deliver a call.

Similarly, Liu and Maguire described a generalised network architecture that incorporated prediction with the goal of supporting mobile computing [13]. Mobile units wirelessly communicating with the network provide updates of their locations and a predictive model is created, allowing services and data to be pre-cached at the most likely future locations.

Davis et al. utilised location modelling in their investigations of highly-partitioned ad-hoc networks [6]. As mobile agents moved around a simulated environment, passing packets between stationary agents, location models were created. The models allowed an agent that was less likely to deliver a particular packet to pass it to an agent that had a higher likelihood of successful delivery.

Unlike those using fixed sensors, systems using GPS to detect location must have some method to determine which locations are significant, and which may be ignored. In their investigations of automatic travel diaries [20], Wolf et al. used stopping time to mark the starting and ending points of trips. In their work on the *comMotion* system [14], Marmasse and Schmandt used the loss of GPS signals to detect buildings. When the GPS signal was lost and then later re-acquired within a certain radius, *comMotion* considered this to be indicative of a building. This approach avoided false detection of buildings when passing through urban canyons or suffering from hardware issues such as battery loss.

2 Applications

Potential applications for a location-modelling system fall into two main categories: single-user, or non-collaborative, and multi-user, or collaborative. Single-user applications are those that can be applied to one person with only her own location model. Collaborative applications, on the other hand, are useful only with two or more location models, and may be used to promote cooperation and collaboration between individuals.

2.1 Single-user applications

In their paper on the *comMotion* system [14], Marmasse and Schmandt explore the idea of an agent that learns frequented locations. The user may associate a to-do list with each location, in the form of text or audio. When the user reaches a location, the applicable to-do list is displayed. One to-do example Marmasse provides is that of reminding the user of her shopping list as she nears a grocery store. If the user was driving, however, reminding her that she needed to visit the store as she passes it could be frustrating and distracting. Instead, reminding the user several miles in advance, or even as she enters her car, would be more productive.

Many other early-reminder applications may easily be imagined. For example, suppose a user has a library book she needs to return. If her location model predicts she'll be near the library later in the day, she can be reminded to take the book on the way out of the house.

Reminders are not the only possible use for the single user: wearable computer systems issues may be addressed as well. Wireless networks, while very useful for wearable users, are often inaccessible due to lack of infrastructure, radio shadows caused by buildings, power requirements and other problems. In some cases, however, this lack of connectivity may be hidden from the user by caching [11]. For example, if a user composes an e-mail while riding the subway, the wearable may add the message to its outgoing queue and wait to send it until the network is available. On the other hand, if the message is urgent, this behaviour may not be appropriate. If the user is predicted to be out of range of the network for some time, she could be alerted of possible alternate travel paths that will allow her message to be sent.

For less urgent e-mail and Internet services, it may be desirable to delay transmission even when a wireless connection is available. Energy is one of the most precious resources for mobile devices, and the amount of energy needed to transmit a message may go up with the fourth power of distance in some situations [3]. In addition, the cost of transmission may vary with the time of day and the type of service that is used. Location prediction abilities could allow a wearable computer to optimise its transmissions based on the cost and availability of service in various locations and the knowledge of how its user moves throughout the day.

2.2 Multi-user applications

When multiple people share their location models, either fully or partially, many useful applications become possible. The models could be shared by giving full or partial copies to trusted associates, delegating the coordination of models to a central service, or allowing remote queries from colleagues whenever information is needed. These three options run from more convenient to more accurate: copying models allows instant access to another person's model, but doesn't guarantee that it will be up to date; a central service allows models to be updated whenever one party has connectivity, making the model more likely to be valid, and remote queries can ensure accuracy at the possible cost of high latency.

Regardless of the sharing mechanism, there are several interesting applications that could be implemented. The simplest scenario is thus: a user, who we'll call Alice,

could ask, "Will I see Bob today?" This type of query gives Alice a useful piece of information—if she needs to bring a thick textbook to Bob, she will only want to take it with her if she's likely to see Bob that day. The query also preserves Bob's privacy, since it is never revealed to Alice when she'll see Bob, where she'll see Bob, or where else Bob has been that day.

One step up from this simple application is the common problem of scheduling a meeting for several people. In his description of the QuickStep platform [16], Jörg Roth described a sample application that facilitated scheduling meetings by showing each user the others' calendars. While a participant would see each user's calendar and when they were unavailable, the labels that showed the reason for the unavailability were removed. Privacy could be preserved to an even greater extent by hiding the schedules themselves from the group and deferring the schedule suggestions to some central system. Such a system could not only find a time when every member is available, but a time when each person is close to the desired meeting location.

Another possibility is encouraging serendipitous meetings between colleagues. Suppose Alice's model indicates that she has lunch at a certain café every Thursday. If Bob happens to be in that general area on Thursday near lunchtime, he could be notified that Alice might be eating nearby so he can give her a call and meet with her.

Michael Terry's Social Net [19] presents an innovative way to meet people with similar interests. It "searches for patterns of physical proximity between people, over time, to infer shared interests between users." Social Net has been implemented using the wireless capabilities of the Cybiko [5] toy to detect proximity. The Cybiko's low (300 foot) range suggests that using location models might be a better way to provide proximity input to Social Net—two people who work in the same building, for example, might be more than 300 feet away from each other almost all of the time. A model that represents places rather than proximity would have a better chance of noticing those people's co-location.

Rather than linking people with similar interests, location modelling could allow otherwise unconnected individuals to exchange favours with each other. For example, suppose Alice needs a book on cryptography for her research, but will not have time to go to the library for several days. She could submit her request to some central arbitration system. The system could look at all of the location models it has for various people, and perhaps discover that Bob will be near the library soon, and not long thereafter near Alice's location. If he is willing, Bob can then pick up the book and deliver it to Alice. One can imagine a sort of reputation system based on favours like these, such as that described in Bruce Stirling's book *Distraction* [18]. In their work on the WALID system [12], Kortuem et al. created a framework for negotiation between wearable computers that explores many of the issues inherent in this sort of system.

The final application for location models we will discuss is that of intelligent interruption. While James Hudson demonstrated that the nature and desirability of interruption is often uncertain [10], there are certain situations in which being interrupted by, say, a ringing cellular phone is definitely *not* acceptable. By allowing one's wearable computer to manage potential interruptions like cell phones, location models can be used to make an intelligent guess about whether the user is interruptible or not.

As an example, imagine that the user has a class from 4:00 to 5:00 p.m. every day. When the user enters the classroom, her wearable, having learned from previous situations, automatically turns her cell phone ringer off. If someone calls during the class, her wearable answers for her, perhaps telling the caller that she will be available around 5:00 when her class is over. As the user walks out of class, her wearable reactivates her phone's ringer and alerts her that someone has called.

3 The pilot study

In order to begin investigating the benefits provided by location modelling, we constructed a system to record and model an individual's travel. In its current form, the system performs modelling and prediction on different scales and allows queries to the model such as "The user is currently at home. What is the most likely place she will go next?" and "How likely is the user to stop at the grocery store on the way home from work?" Combining the answers to these questions for several users can lead to serendipitous meetings: if the response to "Where is the user most likely to go next?" is the same for two colleagues, they can be alerted that they're likely to meet each other.

To date, we have conducted two studies. In 2001, we performed a pilot study with a single user over the course of four months [1]. We used this study to develop our algorithms; in order to show that the system generalises, we conducted a second study in 2002 with six users which lasted seven months. We discuss and compare the results of both studies in the following sections.

3.1 The apparatus

In our original study, we used a Garmin model 35-LVS wearable GPS receiver and a GPS data logger, both from GeoStats [9], to collect data from one user for a period of four months. During those four months, the user travelled mainly in and around Atlanta, Georgia. Our data logger recorded the latitude, longitude, date and time from the GPS receiver at an interval of once per second, but only if the receiver had a valid signal and was moving at one mile an hour or greater. When the receiver was indoors or otherwise had its signal blocked,

the logger therefore did not record anything. Because humans walk at an average of three miles per hour, we captured most forms of transit, including automobile. Figure 1 shows this original data superimposed on a map of Atlanta.

3.2 The methodology

When creating our location modelling system, we wanted to process the data as much as we could without using any a priori knowledge about the world; we hoped to find techniques that would automatically pick out patterns in the data that would mirror what we observe about human movement. We were at least partially successful, but the so-called "Ugly Duckling Theorem" from pattern recognition [7] basically states that there is no "best" set of features; we must give *some* input into our algorithms, and this will eventually influence our results. We are encouraged, however, because the algorithms described in the following sections were developed by looking at our data from the first study, which was for a single user, but they have proved effective on all the subsequent data we have collected.

3.2.1 Finding significant places

In order for any predictions we make to be meaningful, we want to discard as much of the data as possible. It would be quite useless to tell the user, "You're currently at 33.93885° N, 84.33697° W and there's a probability of 74% that you'll move to 33.93885° N, 84.33713° W next." Instead, we would like to find points that have some significance to the user and perform predictions with those.

The most logical way to find points that the user might consider significant is to look at where the user spends her time. It's unlikely that the user would consider somewhere where she never stopped (say the middle of the highway) worth consideration, but quite likely that she would like predictions related to her workplace or home. It also seems likely that, at least for most people, locations that could be considered significant will be inside buildings where GPS signals do not reach. This means that there will be a stream of recorded data until the user enters a building, then a time gap, and then a resumption of data when the user exits the building. We used this idea to find what we call *places*. We define a *place* as *any logged GPS coordinate with an interval of time* t *between it and the previous point*.

In order to decide what value for choose for t, we plotted the number of *places* found for many values of t on a graph (Fig. 2) and looked for an obvious point at which to choose t. Unfortunately, there is no clear point on the graph to choose. In the end, we decided on ten minutes as an amount of stopping time that users might reasonably consider significant. Because of the characteristics of GPS, this is "safer" than choosing smaller values such as one or five minutes because urban canyons and the like can cause signal loss with re-acquisition times of 45 seconds to five minutes [8]. This could lead to erroneous detection of *places* in areas with intermittent signals.

3.2.2 Clustering places into locations

Because multiple GPS measurements taken in the same physical location can vary by as much as 15 meters, the logger will not record exactly the same GPS coordinate for a location even if the user stops for ten minutes at precisely the same point every day. This makes

Fig. 1 All GPS data captured during the four month period of the 2001 study

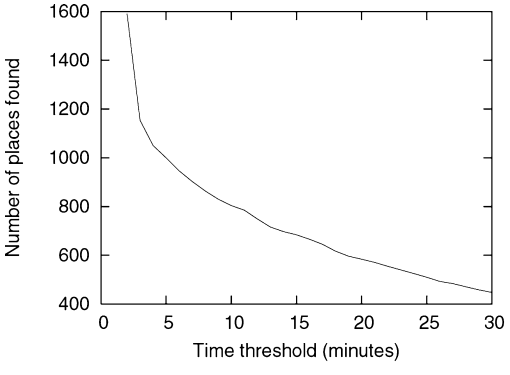

Fig. 2 The number of *places* found for varying values of time threshold t for data from the 2001 study

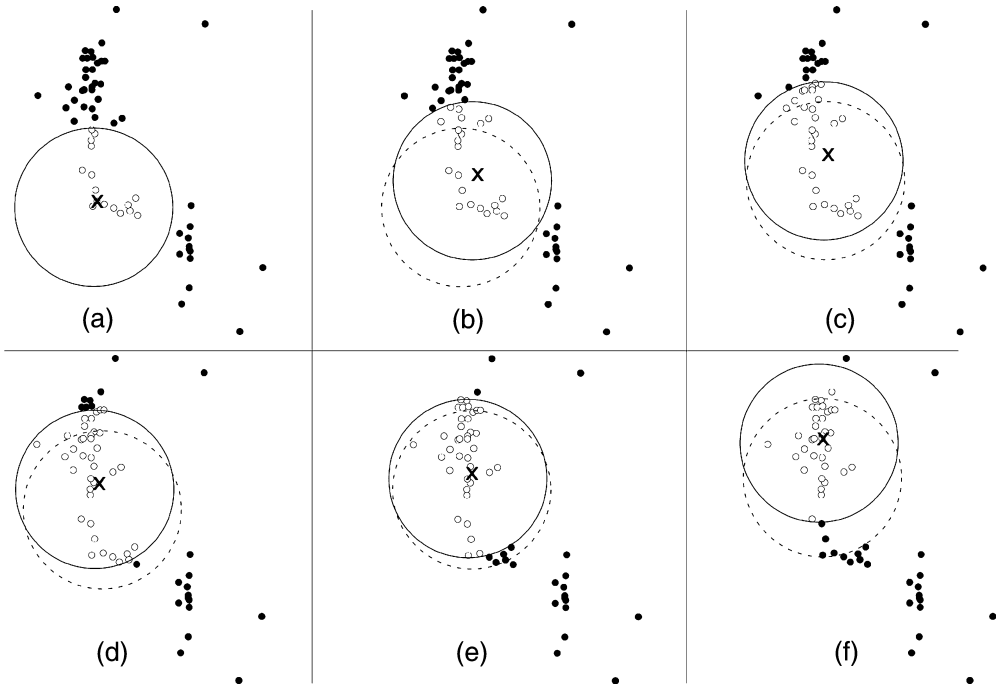

Fig. 3 The illustration of the *location* clustering algorithm. The X denotes the centre of the cluster. The white dots are the points within the cluster, and the dotted line shows the location of the cluster in the previous step. At step **e**, the mean has stopped moving, so all of the white points will be part of this *location*

attempting to use *places* for modelling impractical; we could end up with predictions between two points separated by only a few feet. For this reason, we create clusters of *places* using a variant of the k-means clustering algorithm. We call the resulting clusters *locations*, and use them instead of *places* when forming our models.

The basic idea of our clustering algorithm is to take one *place* point and a radius. All the points within this radius of the *place* are marked, and the mean of these points is found. The mean is then taken as the new centre point, and the process is repeated. This continues until the mean stops changing. When the mean no longer moves, all points within its radius are placed in its cluster and removed from consideration. This cluster is then used as a new location, and has a unique ID assigned to it. The procedure repeats until no *places* remain and we are left with a collection of *locations*. An illustration of this is shown in Fig. 3.

To make our predictions as useful and specific as possible, we want to have *locations* with small radii, thus differentiating between as many distinct *locations* as possible. However, if we make the radii too small, we will end up with only one *place* per *location*. This would create the same problem we were originally trying to solve by clustering! On the other hand, if we make the radii too large, we could end up with unrelated places grouped together, such as home and the grocery store.

To find an optimal radius, we run our clustering algorithm several times with varying radii. We then plot the results on a graph and look for a "knee" (Fig. 4). The knee signifies the radius just before the number of *locations* begins to converge to the number of *places*. In order to find the knee in the curve, we start at the right-hand side of the graph, and work our way leftwards. For each point on the graph, we find the average of it and the next n points on the right. If the current point exceeds the average by some threshold, we use it as the knee point. This method is a simple variant of looking for a significant change in the slope of the graph.

Figure 5 shows the *locations* found for a time threshold of ten minutes and a *location* radius of one half mile. Note the vast reduction in points from the full set of data in Fig. 1—while the user travelled by car and foot over 1,600 miles, there are only a handful of places that the user actually stopped at for any length of time.

3.2.3 Learning sublocations

When creating our *locations* with a particular radius, we may subsume smaller-scale paths—for example, if our radius is chosen to make prediction efficient on a city-wide scale, we may obscure prediction opportunities on a campus-wide scale. Choosing a small radius to allow for

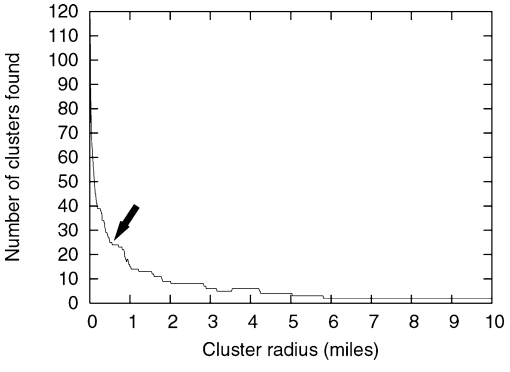

Fig. 4 The number of *locations* found as the cluster radius changes. The arrow denotes a knee in the graph—the radius just before the number of *locations* begins to converge to the number of *places*

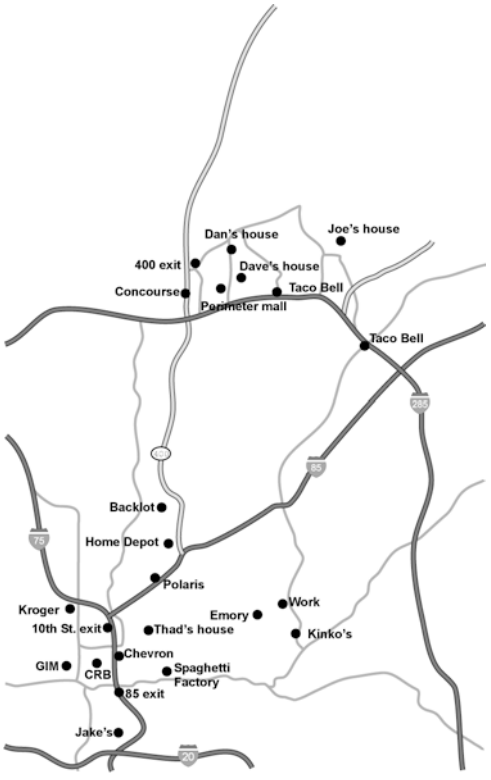

Fig. 5 The *locations* in the Atlanta area, as determined with a time threshold of $t = 10$ minutes and a *location* radius of one half mile

multiple campus *locations*, however, will remove the ability to predict broader trips such as "Campus → Home" in favour of things like "Physics building → Home," "Math building → Home," and so forth.

To solve this problem, we introduce the concept of *sublocations*. For every cluster we find in the main list of *places*, we determine if there is a network of *sublocations* within it that may be exploited. This is accomplished by taking the points within each *location* and running them through the same clustering algorithm described above, including graphing varying radii and looking for the knee in the graph (Fig. 6). If the knee exists, that radius is used to form *sublocations*, which can then have the same

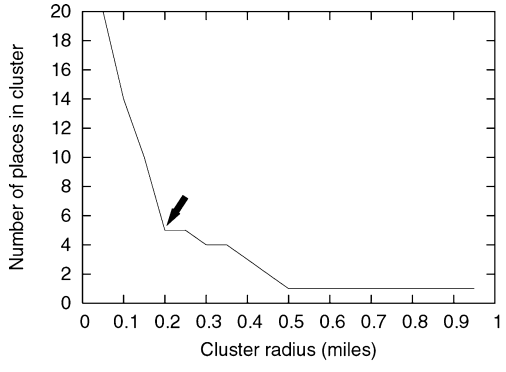

Fig. 6 The number of *places* found in a *location* as the *location's* radius changes. The arrow indicates the knee in the graph; the radius at this point will be used to form *sublocations*

prediction techniques applied as the main *locations*. If no knee exists, we assume that there are not enough points within the *location* to form *sublocations*. An example of *sublocation* creation can be seen in Fig. 12.

The *sublocation* algorithm can be applied multiple times, to create *sub-sublocations* and so on. This can allow for many scales, such as nation-level, city-level, campus-level and so on; given sensors with higher accuracy than GPS, this could even be extended to building- and room-level *sublocations*.

3.2.4 Prediction

Having reduced our original hundreds of thousands of GPS coordinates to just a few significant *locations*, the next step is to create the predictive model we discussed earlier.

Once we have formed *locations* from all of the data, we assign each *location* a unique ID. Then, going back to the original chronological *places* list, we substitute for each *place* the ID of the *location* it belongs to. This gives us a list of *locations* the user visited, in the order that they were visited.

Next, a Markov model is created for each *location*, with transitions to every other *location*. Each node in the Markov model is a *location*, and a transition between two nodes represents the probability of the user travelling between those two *locations*. If the user never travelled between two *locations*, that transition probability is set to zero.

Figure 7 shows a partial Markov model (created from the preliminary data) with three paths—those for "Home", "CRB", and "VA". Although the full model contains many paths, for clarity only transitions between those three *locations* are shown. The labels on the lines between *locations* show the relative probabilities of each transition; for example, seventy-seven trips were made from "CRB" to other *locations*, and of those trips, sixteen were made to "Home". Of three trips made from "VA" to other *places*, one was made to "Home" and one was made to "CRB".

Note that the number of trips made from "VA" are relatively few as compared to those from "Home" and "CRB." It is possible that "VA" is a new *location*, or one that is seldom travelled to by the user. Since there are so few trips, this node should not be used for prediction; however, the number of trips (in both directions) between "Home" and "CRB" *are* significant and *can* be used for prediction. A simple test on whether a path has sufficient evidence for prediction is to compare the path's relative frequency to the probability that the path was taken by chance. For example, the total number of trips taken from "VA" to other locations was 3; this makes the probability that some path was taken by chance 1/3. This is the same probability as the "VA" to "CRB" transition, so this is probably currently not a useful prediction.

Fig. 7 shows a first order Markov model; that is, one in which the probability of the next state is dependent

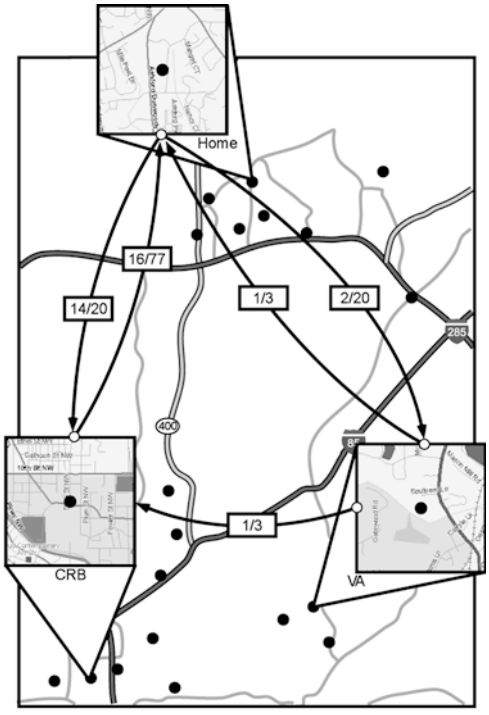

Fig. 7 A partial Markov model of trips made between home, the Centennial Research Building (CRB), and the Department of Veterans Affairs (VA). Because some paths are not shown, the ratios do not sum to 1

only upon the current state. We also have the ability to create *nth* order models, in which the probability of the next state is dependent on the current state and the previous $n-1$ states.

By using higher-order models, we can get significant increases in predictive power. For example, in Table 1 the user's probability of travelling from A to B (''Home'' to ''CRB'') is 70%. However, if we know that the user was already at B, the user's probability of travelling from A to B increases to 81%! Using a second-order model like this could prove particularly useful in cases where the user makes a stop on the way to a final destination, such as stopping by the coffee shop on the way to work. If the system detects the sequence ''Home → Coffee Shop,'' the chances of ''Work'' being the next destination might be higher than when ''Home → Grocery Store'' was detected.

Table 1 Probabilities for transitions in first and second order Markov models from preliminary data. Key: A = ''Home'', B = ''CRB'', and D = ''south of Tech''

Transition	Relative frequency	Probability
$A \rightarrow$ B	14/20	0.7
$A \rightarrow$ B $\rightarrow$ A	3/14	0.2142
$A \rightarrow$ B $\rightarrow$ C	2/14	0.1428
$A \rightarrow$ B $\rightarrow$ D	3/14	0.2142
$A \rightarrow$ B $\rightarrow$ E	1/14	0.0714
$A \rightarrow$ B $\rightarrow$ F	1/14	0.0714
$A \rightarrow$ B $\rightarrow$ G	1/14	0.0714
$A \rightarrow$ B $\rightarrow$ H	1/14	0.0714
$A \rightarrow$ B $\rightarrow$ I	1/14	0.0714
$B \rightarrow$ A	16/77	0.2077
$B \rightarrow$ A $\rightarrow$ B	13/16	0.8125
$B \rightarrow$ A $\rightarrow$ J	3/16	0.1875
$B \rightarrow$ C	10/77	0.1298
$B \rightarrow$ C $\rightarrow$ A	6/10	0.6
$B \rightarrow$ C $\rightarrow$ K	4/10	0.4
$D \rightarrow$ B	5/7	0.7142
$D \rightarrow$ B $\rightarrow$ A	2/5	0.4
$D \rightarrow$ B $\rightarrow$ L	2/5	0.4
$D \rightarrow$ B $\rightarrow$ M	1/5	0.2

The ability to use *nth* order Markov models raises the question of what the appropriate order model is to use for prediction. Bhattacharya and Das have examined this question from an information theoretic standpoint [2]. In practice, a natural limitation is the quantity of data available for analysis; as shown in Table 1, even with four months of data, the number of second order transitions is relatively small. For this reason, we limited ourselves to a second order model in the pilot study.

4 The Zürich study

To determine whether the algorithms developed during the pilot study generalise, we conducted a second study in Zürich, Switzerland with multiple users. Our users had never lived in Zürich before but had moved to the city as a group to conduct research with another university. We equipped the users with GPS systems soon after arrival. By correlating the *locations* across subjects with similar schedules we could establish a sense of whether the results of our *place* and *location* corresponded with areas where the users actually spent their time. By allowing the users to name these locations independently and comparing the results, we can establish whether the locations are meaningful on a more ''semantic'' level (i.e., the users might have independently established an identity for the location for their own purposes in everyday conversation). Finally, by comparing predictions made across similar users, we can begin to determine how the complete algorithm generalises from the pilot study.

4.1 Changes to the apparatus

For the next phase of our study, we wished to collect more data. To this end, we acquired six more GPS receivers and data loggers from GeoStats. The receivers were the same Garmin units we used before (with an accuracy of 15 meters RMS [8]), and the data loggers were updated with more memory, allowing them to log roughly 200,000 GPS coordinates before needing to be cleared. Because of this extra capacity, we elected to turn off the speed-limiting feature of the loggers, in case we had use for the extra data in the future.

Previously, our GPS receiver was powered by six ''D''-size batteries, and the logger was powered by a single 9-volt battery. Because of the number of units we

had, we needed to create a better power solution. We therefore used a single Sony InfoLithium NP-F960 camcorder battery to power both the GPS receiver and the data logger. We added a 5-volt regulator for the receiver, and were able to power the logger directly from battery voltage. Although this reduced the bulk and weight of the setup, it did require each subject to remember to change and charge the battery daily, since the GPS receiver draws between 500 and 600 milliwatts.

We equipped six users with the GPS receivers and loggers during a seven-month research program in Zürich, Switzerland. The users were requested to carry the units with them as much as possible, and were instructed to charge and change the battery daily. This requirement occasionally proved problematic, as some users often forgot to change the battery. Users also often forgot or chose not to wear the receiver and logger. Also, near the beginning of the study one user broke the cabling for his unit and was unable to collect any data for the rest of the study. Despite these problems, we were able to log a total of nearly 800,000 data points.

One issue we discovered while examining the recorded data is that the GPS receivers have a ''dead-reckoning'' feature, which, when the GPS signal is lost, outputs data extrapolated from the last known heading and velocity for 30 seconds. We also found that the GPS receiver would occasionally output a few completely spurious points, possibly when the signal was bad; the most severe example of this is shown in Fig. 8. However, due to the nature of our prediction algorithms (described in a previous section), these glitches had little impact on our eventual results.

After consultation with Garmin technical support, we learned (too late) that dead-reckoning can be turned off, and that the NMEA string returned from the GPS receiver will contain extra information about the signal, such as confidence. Because the current data logger interprets the NMEA string and stores only the latitude, longitude, date and time, we are investigating building our own logger that will save all of the information provided by the receiver. We are also looking into the possibility of using GPS-enabled cellular phones coupled with wearable computers to log data; these would be an advantage in that users would be more likely to carry the units and the batteries would be charged along with the phone.

Fig. 8 An example of spurious GPS data; the user did not spend time in the Mediterranean

4.2 Changes to the methodology

When finding *places*, an important consideration is how time is used—in our previous study, we considered a point a *place* if it had time t between it and the *previous* point. This basically meant that *places* would be detected when the user exited a building and the GPS receiver reacquired a lock. Our current method, however, registers a *place* when the signal is *lost*, and so is not dependent upon signal acquisition time. Figure 9 shows the difference between these two methods. Assuming some knowledge about the general habits of human beings—namely, that they will tend to visit a few places often—it seems apparent that the second method does a much better job at detecting where users spend their time. The results of the second method, in Fig. 9b, also match closely users' actual experiences.

We also revisited the problem of choosing a time threshold with our new data from Zürich. Unfortunately, as shown in Fig. 10, t and the number of *places* found again have a relationship such that there is no obvious point at which to choose a value for t. As such, we kept our original ten-minute threshold for t. Figure 11b shows the *places* found with this ten-minute threshold for one of the users in Zürich.

4.3 Evaluation

To evaluate how the algorithm developed in the pilot study generalises, we present two results. The first result investigates the correlation between the names assigned to particular *locations* by individual users. The second result demonstrates that even in a significantly different environment, the same prediction method generates consistent results across multiple users.

Figure 11 shows sample data for one of the users in the Zürich study, illustrating how the data is significantly reduced by transformation from raw data into *places* and *locations*. Figure12 provides an example of *sublocation* creation from some of the Zürich data.

4.3.1 Naming across users

To help investigate our algorithms, we developed a visualisation application called GPSVis using the free user

Fig. 9 Picture **a** shows the results of the old *place* finding algorithm, while **b** shows the results of the new algorithm on the same data. Clusters are much more evident in **b**, and the clusters match well with users' experiences. Each colour (or shape) of dot in the pictures represents a different user

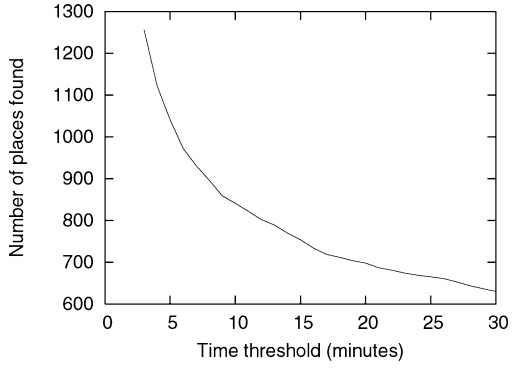

Fig. 10 The number of *places* found for varying values of time *t* for each user in the 2002 study, summed

interface toolkit Gtk+. GPSVis automatically downloads maps from the Internet and displays GPS points, *places* and *locations* on the maps, and allows the user to scroll around and zoom in on any area. It also provides

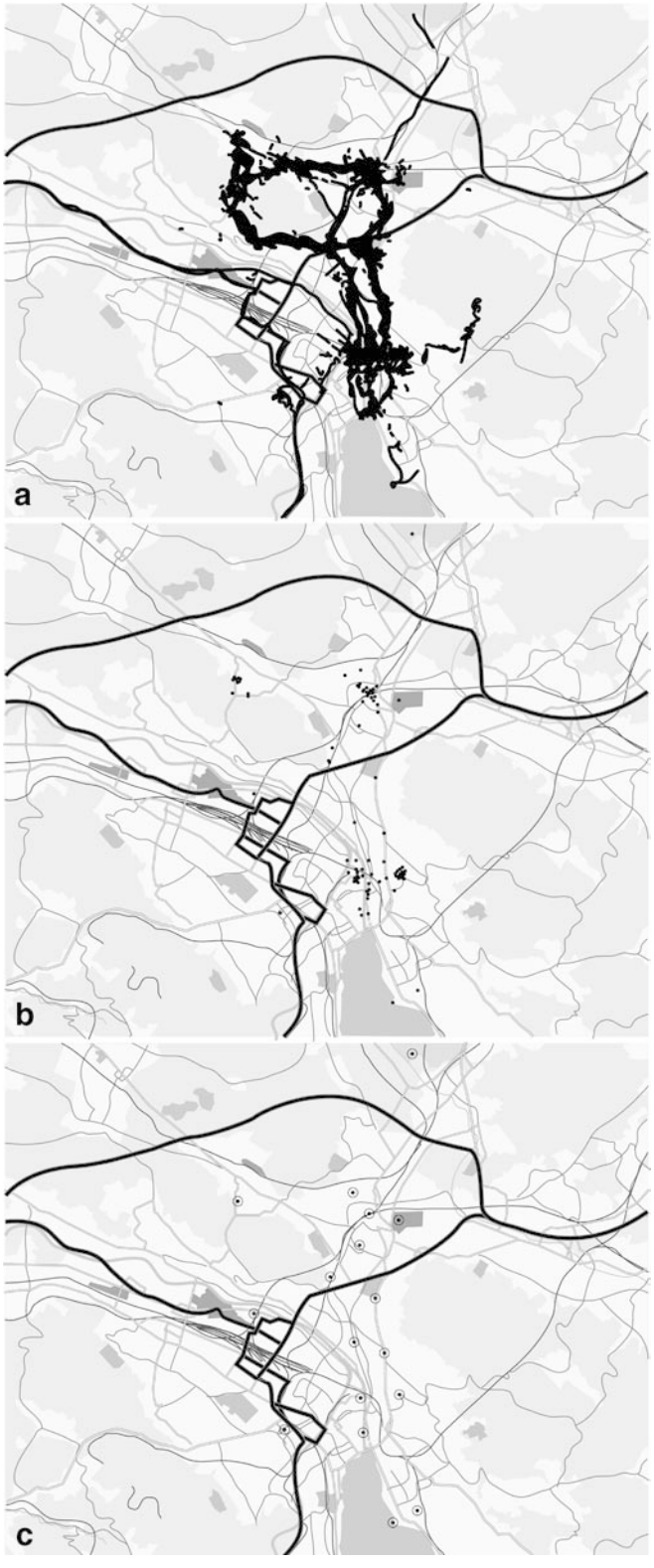

Fig. 11 An illustration of the data reduction that occurs when creating *places* and *locations*. Picture **a** shows the complete set of data collected in Zürich for one user, around 200,000 data points. Picture **b** shows the roughly 100 *places* that were found using this data. In picture **c**, we see that this has been reduced to just 17 significant *locations*

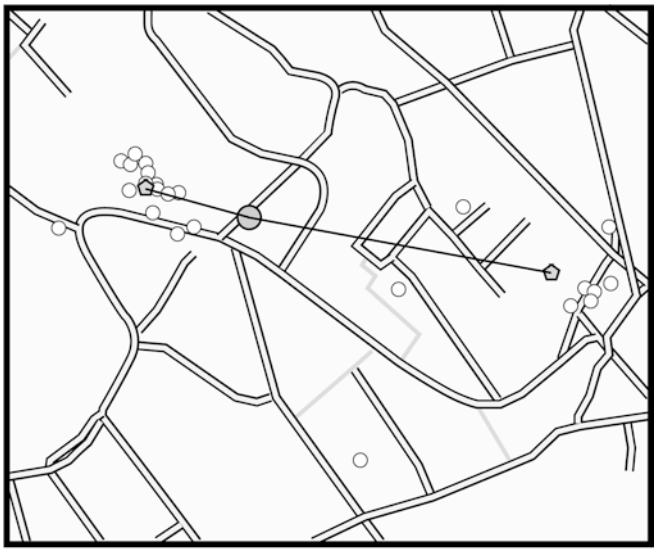

Fig. 12 A single *location* divided into two *sublocations*. The large, green dot is the original *location*, which is centered to the right of the ETZ building in Zürich. The smaller, white dots are the *places* that made up that *location* and the two light blue pentagons are the *sublocations* that were made from the *location*

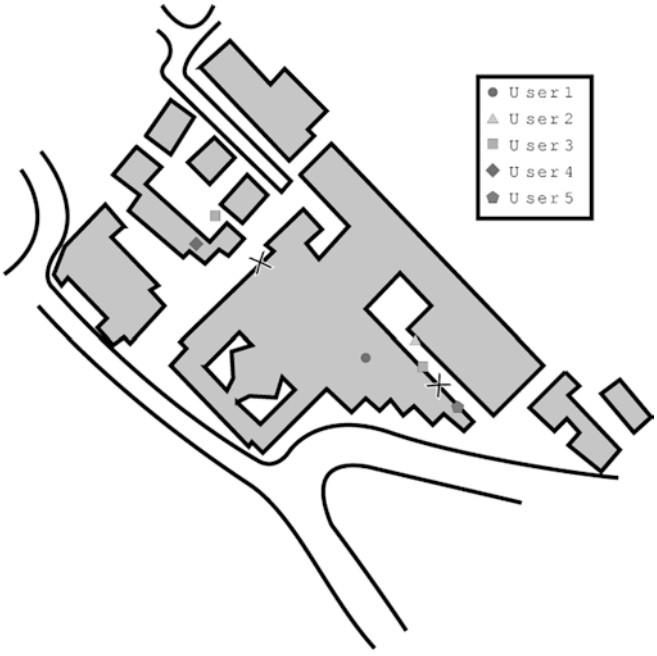

Fig. 13 Locations named "ETZ" by users overlaid on an outline of the actual ETZ building (there are two orange dots because one user named two locations "ETZ"). The Xs correspond to building entrances. The mean distance to the centre point of the six points was 48.4 meters, and the standard deviation was 20.8 meters

the ability to click on any *location* and furnish it with a name or other identifier.

Using GPSVis, we asked each of our subjects to give names to each *location* found for that individual in the city of Zürich. We instructed them to enter "unknown" as the name for any *location* they did not recognise or remember. In order to cut down on spurious *locations*, we did not show *locations* with only one *place* inside them, reasoning that the user had been there only once, if at all.

Of the five users, three had 11 *locations* within Zürich, one had 9 and one had 6. It is interesting (though not necessarily significant) that the three users with more *locations* were in Zürich for the full seven month research program, while the two with 9 and 6 *locations* were in Zürich for only four and three months, respectively. The most "unknown" *locations* any user had in Zürich was 2, and these can be explained by the time lag (about three months) between returning from Zürich and asking the users for names; in fact, several users mentioned the effect of the time lag on their memory.

Although the subjects lived in various places in Zürich, they all worked in the same building (called ETZ). We used this as an opportunity to verify that our *location*-finding algorithms picked common places as frequently-visited locations for everyone. Each person's collection of names included "ETZ," so we mapped those points against a building-level map of Zürich to see how they correlated. The results can be seen in Fig. 13.

4.3.2 Prediction

We applied the same prediction algorithms developed on the Atlanta data (described in a previous section) to the data from Zürich. Tables 2 and 3 show named predictions from this data. Each prediction has its relative frequency compared against random chance, that is, the odds of that path being picked purely by random.

5 Discussion

In our pilot study, we collected four months of data from a single user and then developed algorithms to extract *places* and *locations* from that data. These *locations* were then used to form a predictive model of the

Table 2 Probabilities for transitions for various orders of the Markov model for one of the Zürich users. This user had 215 visits to 18 unique *locations*. Key: A = "ETZ", B = "Home", C = "Sternen Oerlikon tram stop", D = "Zürich Hauptbahnhof", E = "Bucheggplatz tram stop", and F and G are unlabelled (outside of Zürich) GPS coordinates. Random chance was determined by Monte Carlo simulation

Order	Transition	Relative frequency	Random chance
1	$A \rightarrow B$	25/61 = 0.41	0.0065
1	$A \rightarrow D$	16/61 = 0.26	0.0065
1	$A \rightarrow C$	11/61 = 0.18	0.0065
2	$A \rightarrow B \rightarrow C$	17/25 = 0.68	0.0002
2	$A \rightarrow B \rightarrow A$	7/25 = 0.28	0.0002
2	$A \rightarrow B \rightarrow E$	1/25 = 0.04	0.0002
3	$A \rightarrow B \rightarrow C \rightarrow A$	12/17 = 0.71	0.000014
3	$A \rightarrow B \rightarrow C \rightarrow F$	2/17 = 0.12	0.000014
4	$A \rightarrow B \rightarrow C \rightarrow A \rightarrow D$	6/12 = 0.50	0.00000097
4	$A \rightarrow B \rightarrow C \rightarrow A \rightarrow G$	5/12 = 0.42	0.00000097

Table 3 Probabilities for transitions for various orders of the Markov model for a second Zürich user. Key: A = "Home", B = "Kirche Fluntern bus stop", C = "Post office", and D = "Klusplatz tram stop"

Order	Transition	Relative frequency	Random chance
1	$A \rightarrow B$	24/34 = 0.71	0.005
1	$A \rightarrow C$	2/34 = 0.06	0.005
2	$A \rightarrow B \rightarrow A$	11/24 = 0.46	0.0038
2	$A \rightarrow B \rightarrow D$	9/24 = 0.38	0.0038
3	$A \rightarrow B \rightarrow A \rightarrow B$	8/12 = 0.67	0.00376
3	$A \rightarrow B \rightarrow A \rightarrow D$	2/12 = 0.17	0.00376

user's movements. The model demonstrated patterns of movement that occurred much more frequently than chance and were significant in the context of the user's life. These preliminary results suggested that our method may be able to find *locations* that are semantically meaningful to the user.

We next used these algorithms on new data collected from multiple users over a longer time span in a new environment. Locations discovered by the algorithm that were common to several users were named similarly by those users, indicating a certain level of common meaning associated with those locations. In addition, a building known to be common to all the users was automatically labelled as a *location* by the algorithm in each user's data set. The *locations* discovered had a mean distance to the centre of the points of 48.4 meters (approximately the length of the building) and a standard deviation of 20.8 meters. *Locations* shared by a subset of the users were also independently discovered in each user's data. Thus, the algorithm seems to give consistent results across subjects. The predictive models generated for each user based on their *locations* showed relative frequencies significantly greater than chance, also indicating that the method from the pilot study generalises.

6 Future work

While we have location prediction fully functioning, we have not yet implemented time prediction—that is, we can predict where someone will go next, but not when. Our next task will be to extend the Markov model to support time prediction; at the same time, we will investigate how variance in arrival and departure times can indicate the importance of events. For instance, if the user always arrives at a certain location within a 15-minute time period, that location may be more important than one with a one-hour variance.

Detecting "structures" through time may be possible; in the same vein as finding paths that people take, we may be able to find general patterns of routine. For example, looking for very long (6–10 hour) gaps in the data may clue us in to when people are at home sleeping without explicitly coding knowledge of day/night cycles into the algorithms.

One limitation of our approach to the Markov models is that changes in schedule may take a long time to be reflected in the model. For example, a college student might have a model that learned the locations of her classes for an entire semester (16 weeks). When the next semester starts, she may have an entirely different schedule; because in our model each transition is given equal weight, it might take the entire semester for the model to be updated to correctly reflect the new information. One way this could be solved is by weighting updates to the model more heavily; we must be careful, however, to avoid unduly weighting one-time trips.

Currently, our system does not update the user models in real-time; this will become more necessary as we add more users to our system. We plan on not only allowing instant integration of location data, but allowing the users to view their models and give feedback; if the user knows her schedule has changed but that the model has not yet detected this, she can update the model as appropriate.

Speed of travel may provide clues to creating *sublocations*. If a person is detected to be moving very slowly in a particular area, it may be an indicator that they are on foot. *Sublocations* could then be automatically created for that area.

Our planned interface will also include an interface to naming *locations*; once a user has been detected at a possible location more than a couple times, we can prompt the user for a name. With multiple users, the system could suggest names for that *location* that other users had previously used. The user could also indicate that the detected *location* isn't meaningful and it could be ignored for future predictions.

Along with our user interface, we would like to create an application to easily enable favour-trading applications such as those discussed earlier. A simple application would allow users to enter to-do items and associate them with particular *locations*. A software agent for the user's community could then search each person's predictive model to determine which person might be near that location. The to-do application would then show the user a rank-ordered list of members of the community that might be in a position to perform a favour for that user.

Using the algorithms we developed on our original data to process the data from our second study verified that our techniques are valid; however, there is still work to be done in this area. We would like to explore how robust our algorithms are to change; one experiment we intend to perform is to randomly permute our lists of *places* and create *locations*. We can then see how dependent the clustering algorithm is on list ordering. We will also investigate other algorithms to see if any provide more natural clustering.

While in our current procedures we discard much of the data, it may be useful to go back to the original data stream when doing prediction. We may be able to detect situations where the user passes through a *location* but does not stop, or only stops briefly.

Having data for multiple people creates some interesting opportunities for jump-starting prediction. If person A is found to be in several of the same *locations* as person B, it may be possible to use person B's collection of *locations* and predictions as a base set of data for person A. Then as person A continues to collect more data, the *locations* and predictions can be updated appropriately.

7 Conclusions

We have demonstrated how locations of significance can be automatically learned from GPS data at multiple scales. We have also shown a system that can incorporate these locations into a predictive model of the user's movements. In addition, we have described several potential applications of such models, including both single- and multi-user scenarios. Potentially, such methodologies might be extended to other sources of context as well. One day, such predictive models might become an integral part of intelligent wearable agents.

Acknowledgements Many thanks to Jan-Derk Bakker for writing a Monte Carlo simulator. Thanks to Graham Coleman for writing visualisation tools and to MapBlast (http://www.mapblast.com) for having freely available maps. Funding for this project has been provided in part by NSF career grant number 0093291.

References

1. Ashbrook D, Starner S (2002) Learning significant locations and predicting user movement with GPS. In: Proceedings of the 6th IEEE International Symposium on Wearable Computers, Seattle, WA, 7–10 October 2002
2. Bhattacharya A, Das SK (1999) LeZi-update: an information-theoretic approach to track mobile users in PCS networks. In: Proceedings of the International Conference on Mobile Computing and Networking, Seattle, WA, August 1999
3. Chang J-H, Tassiulas L (2000) Energy conserving routing in wireless ad-hoc networks. INFOCOM (1):22–31
4. Csinger A (1995) User models for intent-based authoring. Dissertation, The University of British Columbia
5. Cybiko, Inc. http: //www.cybiko.com.
6. Davis JA, Fagg AH and Levine BN (2001). Wearable computers as packet transport mechanisms in highly-partitioned ad-hoc networks. In: Proceedings of the 5th IEEE International Symposium on Wearable Computers, Zürich, Switzerland, 8–9 October 2001
7. Duda RO, Hart PE and Stork DG (2001) Pattern classification. Wiley, New York
8. Garmin, Inc. GPS 35 LP TracPak GPS smart antenna technical specification. http://www.garmin.com/products/gps35/.
9. GeoStats, Inc. http: //www.geostats.com.
10. Hudson JM, Christensen J, Kellogg WA and Erickson T (2002) I'd be overwhelmed, but it's just one more thing to do: availability and interruption in research management. In: Proceedings of the Human Factors in Computing Systems Conference (CHI 2002), Minneapolis, MN, 20–25 April 2002
11. Kistler J, Satyanarayanan M (1992) Disconnected operation in the coda file system. ACM Trans Comp Sys 10(1):3–25
12. Kortuem G, Schneider J, Suruda J, Fickas S, and Segall Z (1999) When cyborgs meet: building communities of cooperative wearable agents. In: Proceedings of the 3rd IEEE International Symposium on Wearable Computers, San Francisco, CA, 18–19 October 1999
13. Liu GY, Maguire GQ (1995) Efficient mobility management support for wireless data services. In: Proceedings of the 45th IEEE Vehicular Technology Conference, Chicago, IL, July 1995
14. Marmasse N, Schmandt C (2000) Location-aware information delivery with ComMotion. In: Proceedings of the Second International Symposium on Handheld and Ubiquitous Computing (HUC), Bristol, UK, 25–27 September 2000
15. Orwant J (1993) Doppelgänger goes to school: machine learning for user modeling. Master's thesis, Massachusetts Institute of Technology
16. Roth J, Unger C (2000) Using handheld devices in synchronous collaborative scenarios. In: Proceedings of the Second International Symposium on Handheld and Ubiquitous Computing (HUC), Bristol, UK, 25–27 September 2000
17. Sparacino F (2002) The museum wearable: real-time sensor-driven understanding of visitors' interests for personalized visually-augmented museum experiences. In: Proceedings of Museums and the Web, Boston, MA, April 2002
18. Stirling B (1998) Distraction. Spectra, Pittsburgh, PA
19. Terry M, Mynatt ED, Ryall K and Leigh D (2002) Social net: using patterns of physical proximity over time to infer shared interests. In: Proceedings of the Human Factors in Computing Systems Conference (CHI 2002), Minneapolis, MN, 20–25 April 2002
20. Wolf J, Guensler R and Bachman W (2001) Elimination of the travel diary: an experiment to derive trip purpose from GPS travel data. In: Proceedings from the Transportation Research Board 80th annual meeting, Washington, DC, 7–11 January 2001