A User-Centered Location Model

Natalia Marmasse and Chris Schmandt

MIT Media Laboratory, Cambridge, MA, USA

Abstract: This paper discusses the user-centered location model used in *comMotion*. In this context, the location model refers to a set of learned places (destinations), which coincide to a latitude and a longitude, that the user has categorized. It also includes knowledge of the routes between the destinations and the time it takes to travel them. The model is based on user experience, i.e. his patterns of mobility, so no two models are the same. We also discuss the pattern recognition models implemented for route learning, route prediction and estimation of time to arrival.

Keywords: GPS; Hidden Markov model, Histogram modeling; Location-aware; Location model; User model

1. Motivation

Wireless communication devices are becoming increasingly ubiquitous and, judging by the number of devices people are willing to carry around, it is clear that there is a desire to be able to communicate from just about anywhere. Since information can be delivered to the user at just about any time, chances are that the user will be performing some other task at that moment. Therefore, applications for such devices should be context-sensitive and able to progressively adapt to the user.

comMotion is a location-aware computing environment which links personal information to locations in its user's life. It provides just-in-time information delivery, such as reminding one of an important meeting on the way to work, or the need to buy milk as one is about to drive by the grocery store on the trip home. For such a system, it is crucial to know where the user is, in order to trigger the relevant information, and it is useful to be able to predict his destination and expected time to arrival.

The user interface is critical for systems which are meant to be always on and available – *comMotion* has both graphical and speech interfaces to its core set of functions. In other publications, we have given details of the user interface, the benefits and limitations of a speech-interface, tradeoffs of wanting to be continuously connected yet having to minimize interruptions, etc. [1,2]. Here we will concentrate more on the location model, its advantages and disadvantages, and on the pattern recognition models that can be used to learn and predict where the user is going and estimate his time to arrival.

2. Overview of *comMotion*

As the user goes about his daily routines, a location learning agent, using the Global Positioning System (GPS), monitors his travel patterns and learns his frequented locations. The premise is that if a user frequents a location often enough, it must be of some importance in his life. Once a new location has been identified as a salient one, the user is prompted to name it – naming the place indicates to the system that it is of importance. At this point the geographic location is converted to a virtual one (such as 'work' or 'home'), and a to-do list is immediately associated with it. Since the system incrementally learns the user's frequented places, no initial configuration is necessary. As the user's routines change, the system will adapt and incorporate the new places.

If the GPS coordinates correspond to an already learned location, it is translated to the virtual location equivalent (such as 'work') and passed on to the message engine, where the existence of relevant pending messages is checked. Reminders, to-do lists, email messages and Web content can be delivered to the different geographic locations. These are triggered when the user is in the appropriate context: physical location, date and time.

Personal and Ubiquitous Computing (2002) 6:318–321

Most previous location-aware applications have used predefined content and/or predefined locations. Such as C-Map [3], CyberGuide [4], Metronaut [5] and City Guide [6]. In the StickePad project [7], neither the locations nor the content are predefined, however the content simply relates to a geographic location where the observations where taken. Back Seat Driver [8] had an inertial navigation system and a concept of a route. It did not learn the routes, rather, given a departure point and a destination, it would figure out a route and by means of a speech interface give the driver directions. It also had a model of the time it took to speak the relevant instructions, and would adapt its commands based on the user's travel speed. *comMotion* can have predefined content associated to locations, however its main feature is user-defined dynamic content and the possibility to subscribe to Web content based on location. As far as we know, no other system observes the user's mobility data to independently learn the frequented locations.

3. Location Model

In previous work we have shown how end-points of routes, that is the destinations themselves, can be learned, provided they are buildings. When GPS signal is lost within a certain radius, and the user later 'reappears' within the same radius, the location is inferred to be a building; since most buildings are GPS opaque. It would also be possible to learn stationery points, for example, of a parked car in a mall parking lot. GPS is a great position sensor for outdoors location – it is global, relatively cheap and maintenance free. However, GPS does have many limitations – signal acquiring times, shadowing from buildings in the so-called urban canyons, and lack of position accuracy due to geometry of visible satellites. When a user is identified a given number of times within a defined radius of an unlabeled location that is understood to be a building, he is prompted for a location name. A radius around that position becomes a labeled destination. A route is defined as the trajectory taken between labeled destinations. There could be multiple routes between the same two destinations, and route AB is not the same as route BA.

The location model includes two major components:

- user-centric salient locations (destinations). These are virtual locations that map to a latitude/longitude and are classified by the user.
- routes between the locations. These are made up of latitude, longitude and time.

The frequented locations are labeled by the user, as opposed to being classified with the help of an external database. The underlying assumption is that places often frequented by a user must have some importance in his life, however, the user will only label locations he considers relevant. For example, a bus stop might be visited often, but a user would not necessarily mark it as a place where specific information should be delivered; although he might want the headline news delivered then. An external database, no matter how extensive, will not always include locations of great importance to the user, for example, Grandma's house. There are also privacy issues with querying an external database for the classification of a certain location associated with a user.

Routes are important to predict where the user is going – he could be alerted of things he has to do on an alternate route, which leads to the same destination. Being able to predict the user's destination enables estimating his time to arrival. The time taken to travel a subsection of a route is variable; it is dependent both on the time of the day, and on the type of road. Knowing the time taken to traverse different subsections of a route can be used to better alert the user (for example presenting a reminder earlier on if traveling on a fast road). The time attribute also enables calculating the current travel time versus the typical travel time. Dead zones, i.e. where GPS signal is lost, could also be modeled and learned, enabling loss of signal to be predicted as well as where and when the user is likely to 'reappear'. If a signal is not regained in the expected time, the probability of still being on the same route would decrease as a function of time. If dead zones are not modeled, we simply know where the user was last seen, and his velocity.

The advantages of this location model are:

- no clutter, since labeled locations are only those of importance to the user,
- the learning of the locations is incremental and adaptive,
- no external database is necessary,

- the data is secure since it resides solely on the client side.

The limitations are:

- it requires time to learn;
- inability to separate locations close to one another, e.g. multiple stores in the same mall would not be distinguished as individual locations. In a huge parking lot, if parked each time at different ends, the system would not associate the different geographic locations to the same virtual place, i.e. 'the mall';
- it is confined to learning outdoor locations;
- shadowing of the GPS signal from tall buildings can be a problem, especially when walking. These dead zones could also be modeled and learned.

4. Route Learning

We used GPS data corresponding to five different routes. The goal was to try different pattern recognition and analysis techniques to classify them and, based on the likelihood of being on a particular route, predict where the user is going and estimate time to destination. Three different methods/models were implemented. Although the testing of the data was not done in real time, rather simulated, it is clear that it could be integrated into the *comMotion* system and done while the user is on-the-move.

4.1. Methods

Though GPS is multidimensional data, the only features used were latitude, longitude and time. Once the different models were trained, they were tested in the following manner: each test route was divided into subsets of incremental sizes corresponding to progression along the route. For example, the first subset included the first five data points of the set, the next included the first five data points plus another five, and so forth. As the system is presented with more data of the trajectory, the destination becomes clearer and the predicted time to arrival adjusted.

Bayes Classifier. Each route is modeled with a Gaussian probability density function. As suspected, the density functions of certain routes were very similar, however routes such as AB and BA, although modeled almost identically, can be disambiguated by knowing which location the user departed from. Once the currently traveled route has been classified, it is possible to calculate how far along the user is by a K-Nearest Neighbor calculation with the training data of the specific route, and hence estimate the time to arrival. However, the time estimates were done as for the histograms.

Histogram Modeling. The training stage results in two histograms (latitude and longitude) which serve to represent the specific route, as well as a table with the average time to destination from each bin. During the testing stage, each subset of the route is divided into bins and compared to the representing histograms. The route is classified based on correlation of both the latitude and longitude histograms. Using this method, along with prior knowledge of the frequency different routes are traveled, it is possible to state, for example, that, when seen leaving home, there is an 80% chance that the user is on his way to the store and a 20% chance that he is going to the post office. The estimated time to destination for each case can easily be given by simply looking up the average time from the last bin of the trajectory subset.

Hidden Markov Model. A left-right, nine-state, two-dimensional continuous HMM was trained on the data of each type of route. The first and last states were fixed to correspond to the end points (labeled 'destinations') of the data. The training set data was perturbed in order to get better results when testing. At each state the HMM outputs a mean (latitude, longitude) and covariance matrix corresponding to a Gaussian distribution.

4.2. Results

Of the three different techniques implemented, the overall best results were obtained from the histograms. Had more training data been available, it is probable that the performance of the HMM would have increased. The histogram modeling not only does well with little training data, but it also makes the task of estimating time of arrival very simple – no extra computation is needed, as would be the case with the other two models.

Although the data was not tested in real time, the method used indicates that it would work while on-the-move, since the division into subsets of data corresponds to progress made along a route.

5. Conclusions

People want to be on-the-move and connected. The very nature of mobility implies that the user will typically be performing another task when he receives information on his mobile communication device. Therefore, effective applications for such devices should not only have appropriate interfaces for hands- and/or eyes-busy situations, but they should also be context-sensitive and able to adapt to the user. We present the user-centric location model used in *comMotion*. This model includes a set of locations (destinations) which are learned by observing the user's patterns of mobility; these locations map to a latitude/longitude and are categorized by the user. In addition, the model includes knowledge of the routes traveled between the different destinations, and the time it takes to travel them. We also discuss different pattern recognition techniques used for route learning, route prediction and estimation of time of arrival.

Although the location model has certain limitations, some of these can be solved. For example, the system takes time to learn locations that are not often frequented. If time is critical, the user can always actively teach the system a location (when *in situ*) by simply pressing a button and naming the place. Alternatively, he can enter the latitude/longitude. Shadowing of the GPS signal by tall buildings can be a problem, especially when walking in urban areas. These dead zones could also be modeled, enabling the system to predict them, as well as when and where the user is expected to 'reappear'.

The location model used in *comMotion* could be integrated into other context-aware mobile systems.

References

1. Marmasse N, Schmandt C. Location-aware information delivery with comMotion. Proceedings Second International Symposium on Handheld and Ubiquitous Computing (HUC), Bristol, UK 2000
2. Schmandt C, Marmasse N, Marti S, Sawhney N, Wheeler S. Everywhere messaging. IBM Systems Journal 2000; 39(3/4)
3. Sumi Y, Etani T, Fels S, Simone N, Kobayashi K, Mase K. C-MAP: Building a context-aware mobile assistant for exhibition tours. Social Interaction and Communityware, Japan, June 1998
4. Long S, Aust D, Abowd C, Atkeson C. Cyberguide: Prototyping context-aware mobile applications. Proceedings Conference on Human Factors in Computing Systems 1996
5. Smailagic A, Martin R. Metronaut: A wearable computer with sensing and global communication capabilities. Proceedings of the International Symposium on Wearable Computing, IEEE 1997
6. Kreller B, Carrega D, Shankar J, Salmon P, Bottger B, Kassing T. A mobile-aware city guide application. ACTS Mobile Communication Summit, Rhodos, Greece 1998
7. Pascoe J, Ryan N, Morse D. Issues in developing context-aware computing. Proceedings First International Symposium on Handheld and Ubiquitous Computing (HUC'99), Germany 1999
8. Davis JR, Schmandt C. The back seat driver: real time spoken driving instructions. Vehicle Navigation and Information Systems 1989

Correspondence to: Ms N. Marmasse, MIT Media Laboratory, 20 Ames Street, Cambridge, MA 02139, USA. Email: nmarmas@media.mit.edu