Location Sensing and Privacy in a Context Aware Computing Environment

Asim Smailagic, Daniel P. Siewiorek, Joshua Anhalt, David Kogan, Yang Wang
Carnegie Mellon University
Institute for Complex Engineered Systems
Pittsburgh, Pennsylvania 15213
Contact: asim@cs.cmu.edu

Abstract- This paper presents and evaluates two location sensing algorithms that we have developed and demonstrated. We present comparative accuracy results, complexity of training the system, and total power consumption required to perform scanning. Our method reduces training complexity by a factor of eight, and yields noticeable better accuracy. The paper also introduces a location information privacy model and reports on user study results. Our results indicate that users expect two unique behaviors from the privacy system, an introvert model where privacy is preferred, and an extrovert model where availability of information is preferred.
Keywords - Location Awareness, Context Aware Computing, Pervasive Computing, Privacy

1. Introduction

Pervasive computing is an emerging field in computer systems research. This new paradigm introduces smart spaces which interact naturally with the user. For these interactions to occur, the system must be aware where the user is. Since location is a very important component in context aware computing, our efforts have focused on developing a Location Service. Our implementation is based on signal strength and access point information from the IEEE 802.11b Wavelan wireless network that covers the entire Carnegie Mellon campus [5]. This is in contrast to a GPS-based approach, which tends to have poor indoor accuracy and also requires each client to be equipped with a GPS unit that adds weight and consumes power. Our goal is to develop a power efficient, scalable and private Location Service that can be used by a variety of clients including wearable and laptop computers.

We have explored two distinct approaches: client-centric and server-centric. In the client-centric approach, the signal strengths of multiple Wavelan access points are obtained by a client. We have implemented several methods of mapping signal strengths to physical locations. Our experiments so far have yielded an accuracy of up to five feet in 70% of our measurements. The server-centric approach is less accurate, but much simpler. In this approach, the identity of the access point currently servicing a client is recorded on a central server and made available to other clients. This information is updated whenever the client moves between access points. This provides a cheap and scalable location service, but one whose resolution is relatively coarse - a sphere that is about 75 feet in diameter, centered on each access point. This approach also raises questions of privacy since anyone can obtain the location of a client to this resolution.

In this paper we describe the client-centric approach based upon a combined triangulation, mapping and interpolation algorithm (CMU-TMI). We evaluate the characteristics of this approach with our previous algorithm, CMU Pattern Matching (CMU-PM), as well as Victor Bahl's RADAR system [1][2]. This paper also introduces attributes of a location sensing system that describe their relative merits for evaluation, accuracy, complexity of training and power consumption. We proceed with outlining the importance of privacy for location information, user study for our location privacy model, and design and implementation of the model. We are using location sensing in our context aware system called Portable Help Desk (PHD).

In addition to RADAR, previous work describes two other methods of implementing location sensing systems. Proprietary infrastructure based systems such as GPS [7], RF and ultrasound [8] provide location information with a high infrastructure cost. Also, proprietary non-infrastructure based systems such as video based systems utilizing pattern matching techniques exists as described by [3] and [4]. These systems currently need significant amounts of training to achieve accurate results.

2. Approach and Architecture

We have developed a five layer architecture for pervasive computer, Handy Andy as shown in Figure 1. The bottom of the figure has a range of mobile and fixed devices. They are not required to be homogeneous in hardware architecture or operating system. The second layer contains proxies for every device. This proxy intercepts user's requests, passes them through a series of user specified filters and

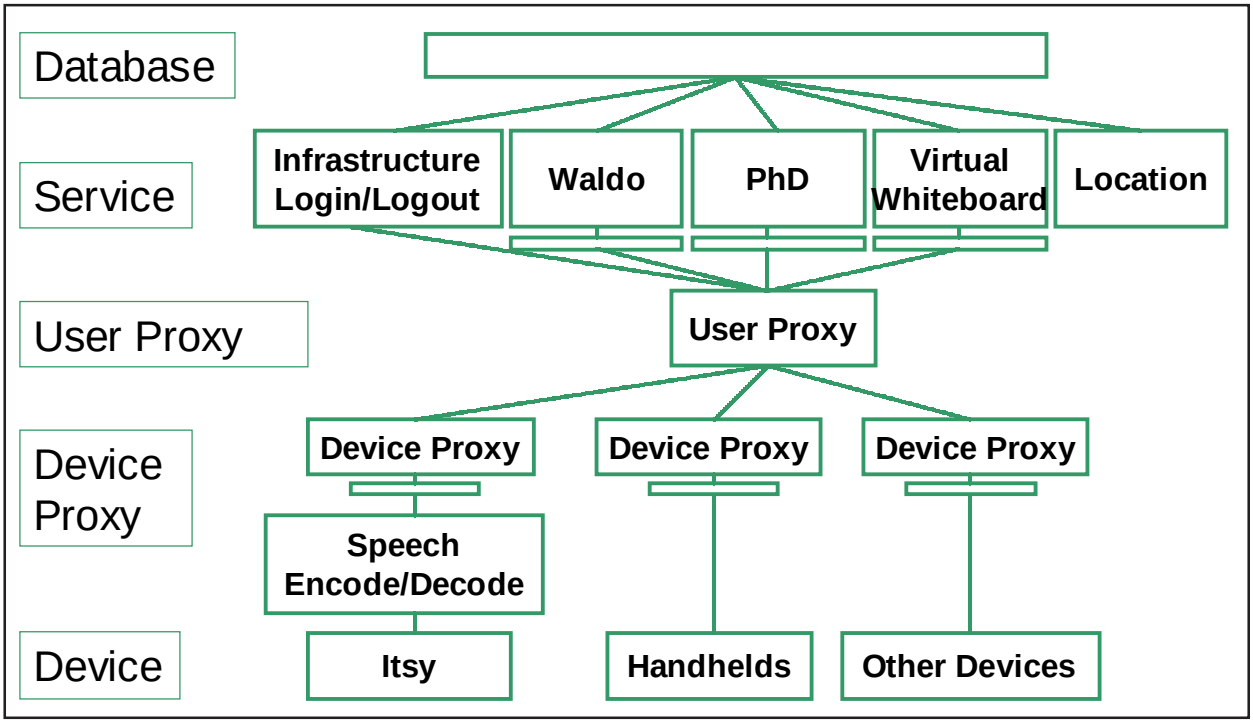

Figure 1 - Handy Andy Pervasive Computing Architecture

forwards the transcoded requests and responses. If the communications channel between the device proxy and the device is broken, the device proxy caches updates until the connection is reestablished. The third layer is the user proxy layer. Every user has their personal user proxy. Applications and a user's state can be stored in this layer. The fourth layer represents the services. Shared applications, utilities and servers are implemented here. The user proxy receives a request, starts an application or forwards it to a service. If the requested service is not well known or defined in the user's preferences, the user proxy invokes the Service Location Protocol to locate the requested service. The fifth layer implements a unified database as a service. All services and user proxies are granted access to the database based on the privileges of the user authenticated to them. This prevents common data such as user name, address and contact information to be duplicated across systems. All requests between layers are made in hypertext transfer protocol, HTTP. Data structures such as an integer, character and string may be sent and received in these requests. Each request includes user identification and device identification.

The client requesting the location of a target user sends their request to a server. The server may use a caching mechanism to answer the request, or send the request to the target user. The target user's computer determines its location and sends the results to the server. The server completes the transaction by sending the location of the target to the client.

3. Portable Help Desk: A Context Aware System

Portable Help Desk (PHD) allows a user to determine the location of other users on campus as well as information about them. It also provides other services such as notifying the user of the closest available printer or where food might be available.

PHD allows a mobile user to build maps of their immediate area, including static and dynamic resources and the location of their colleagues, contact information and resources availability. While tracking a colleague, their contact information is displayed. Printer queues, restaurant hours and stock of carbonated beverages and food in connected vending machines can be displayed. The PHD application is a spatially aware system. Figure 2 illustrates a visual user interface for the PHD application. People and resources are selected in the left pane, the results of the queries are presented in the middle pane while locations of people and resources are displayed in the right pane.

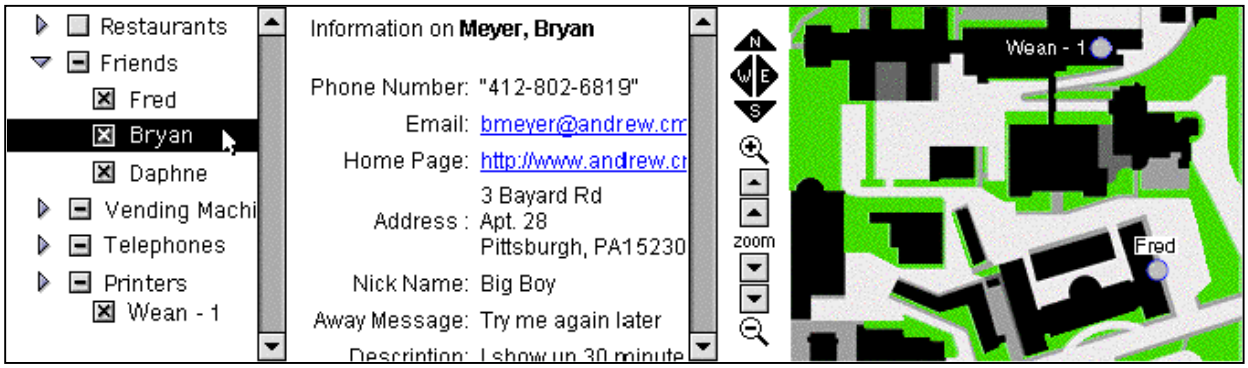

Figure 2 - Portable Help Desk Screen Shot

4. Location Sensing

In this section we describe two location sensing algorithms, CMU's Pattern Matching (CMU-PM), CMU's Triangulation Mapping Interpolation (CMU-TMI), and compare them with two other location sensing algorithms Victor Bahl's RADAR [1][2], and Nearest Basestation. CMU-PM implements a pattern matching algorithm. CMU-TMI implements Triangulation, Mapping and Interpolation. RADAR is a pattern matching algorithm. Nearest Basestation establishes a devices location by determining which basestation it is connected to.

In this paper we do not address the server side implementation. The Lucent Wavelan access points cache the MAC addresses of the client's radios. This cache is persistent, even after the client has left the wireless network. This "ghosting" of MAC addresses prevents accurate location information from being obtained directly from the access points.

Each algorithm requires initial information about the space it is mapping. The RADAR, CMU-PM and CMU-TMI algorithms require training points in order to map signal strengths to a physical location. A single training point is acquired by carrying a device to a physical location and taking measurements of the signal strengths from the access points. The number of training points required by each algorithm varies, and is discussed in Section 4.

The distribution of noise was evaluated. A stationary Wavelan card was set to take measurements for 5 hours with frequency of one sample every 5 seconds [6]. Figure 3 shows the histogram of noise from these measurements. Each dot represents a single potential location. With the largest distribution centered at the actual location of the device. Long term consistency of signal strength information is shown by the low standard deviation. Small's data showed that short term measurements varied significantly [6]. This suggests that multiple measurements must be taken to compensate for short term variation in noise.

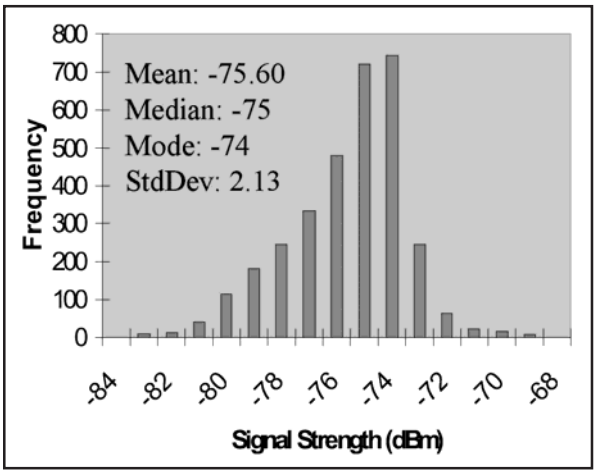

Figure 3 - Stationary Signal Strength Distribution

4.1 CMU-PM

The pattern matching algorithm compares measured signal strengths from nearby access points to recorded training signal strengths to determine the location of a user. For every location, there is a unique reading of signal strengths gathered from a group of access points. A minimum of one access point is necessary to determine location, with the addition of extra access points increasing accuracy. Using fewer than three access points can potentially give an ambiguous answer. The addition of a third access point would allow for an unambiguous reading for the user's location.

For pattern matching determination, samples must be taken in the area for which users will have the need to know their location. For this training, the user's location is manually input into the computer and about 17 samples are taken and averaged. A table is generated, recording what signal levels to expect at different locations. This only has to be done once and can be saved for use in later sessions and on other platforms.

During use, measured values are compared to those in the table and differences are computed. The entry with the smallest difference is taken to be the current position.

4.2 CMU-TMI

The CMU-TMI system requires an investment of initial data. The physical position of all the access points in the area needs to be known. A function is required to map signal strength onto distances. This function was chosen empirically from observations. A set of training points are gathered to map the signal space. An offset vector is generated at each trained position, which permits the correct mapping later on. Interpolation of this training data allows this algorithm to use significantly fewer training points verses the pattern matching algorithms.

4.2.1 Scanning

The device scans all the access points within range to determine their signal and noise levels. The number of visible access points varies depending on location. An average of five access points is sufficient for mapping a location in three dimensions. The algorithm for scanning access points uses previous, well-developed work, accessing the wireless card hardware. Five scans are performed and averaged to reduce the variance in signal strength due to noise.

4.2.2 Triangulation

The signal strengths are used to infer the distance between the client and the access points. Empirical measurements were used to generate the approximate relationship between signal strength and distance. The following equation, where d = distance, s = signal strength, describes this relation:

$$d = .0163 * s^2 - 2.3\ s + 80$$

Contours are generated around each access point, and intersections between contours are found. Because of the inconsistencies of signal space, and the effects of noise, contours from two access points do not necessarily intersect. Two non-intersecting contours are grown or shrunk to estimate an intersection. Noise and the non-linearity of single space cause the intersections to be scattered across a large area. The distance between pairs of intersections are compared to find the most likely location of the device, generating a cluster of possible positions. Figure 4 illustrates the large distribution of calculated possible positions taken from a single location. These measurements include weak signal strengths from access points that are very far away. These weak signal strengths do not contribute to the accuracy of the calculated position, and tend to add more noise. Using only the five strongest access points results in possible position data with significantly less noise. Figure 5 shows the cleaner distribution of points when these distant weak access points are removed. The noise in the signal strengths shown in Section 4 causes the possible location points to be distributed around the actual location. The cluster of possible locations is then averaged to determine the signal space position of the device.

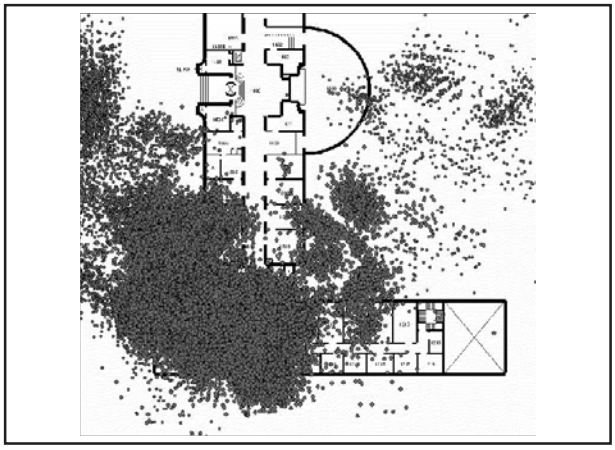

Figure 4 - CMU-TMI Possible Locations All Access Points

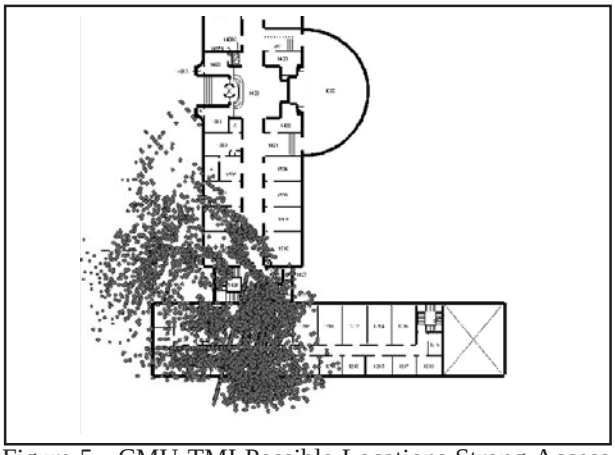

Figure 5 - CMU-TMI Possible Locations Strong Access Points

4.2.3 Mapping

When the signal space position is generated, the nearest set of mappings from signal space to physical space are found. A weighted average based on proximity is applied to the signal space position to calculate the physical space position. This allows an updated physical position to be generated after every signal strength scan.

4.2.4 Smoothing

Noise and the finite granularity of a mapped physical location causes a resulting position to jump with consecutive scans, regardless of whether the device is moving. To alleviate this jitter, and to minimize the error caused by a single poor calculation, the result of previous calculations are averaged with the new calculation. This time averaged location converges towards the new data. The speed of the convergence changes proportional to the distance between the previous value and the new calculated point.

Motion is minimized while the device is still, but still permits the value to react quickly to actual motion. Figure 6 shows sample output from this algorithm. The left most point is the signal space location. The right most point is the computed physical location. The middle point represents the time averaged smoothed location.

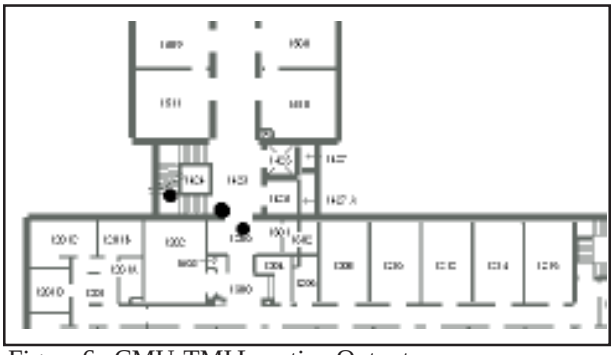

Figure 6 - CMU-TMI Location Output

5. Results and Evaluation

These location sensing techniques each have distinct advantages, likewise, they have their weaknesses. Figure 7 shows the cumulative distribution function verses accuracy for each method discussed. Accuracy of the nearest basestation approach is significantly lower than any other method. Pattern Matching appears to vary based on implementation and testing areas, as the CMU algorithm resulted in the highest accuracy, and RADAR resulted in a slightly lower accuracy. The CMU-TMI algorithm generates results between the two pattern matching accuracies.

Location sensing algorithms have not addressed the complexity of training. CMU's pattern matching, and RADAR have been shown to produce accurate results in a small test area. If the size of the area used and the amount of required training is scaled to a university campus, a large amount of training is needed to maintain the same accuracy. Figure 8 shows the number of training samples to achieve similar accuracies. These figures are linear extrapolations of the reported training point densities from the CMU algorithms and RADAR. Nearest basestation requires no field training for any size building. CMU-TMI results indicate that 1500 points are needed for an area the size of the CMU campus. RADAR requires 12,000 points for the same area, and the CMU pattern matching algorithm requires 28,000 training points for similar accuracy in the same area.

Location sensing is primarily useful for mobile users who have significant power constraints. The amount of power consumed by a given location algorithm affects its usefulness. Battery life for an HP Jornada 680 handheld com-

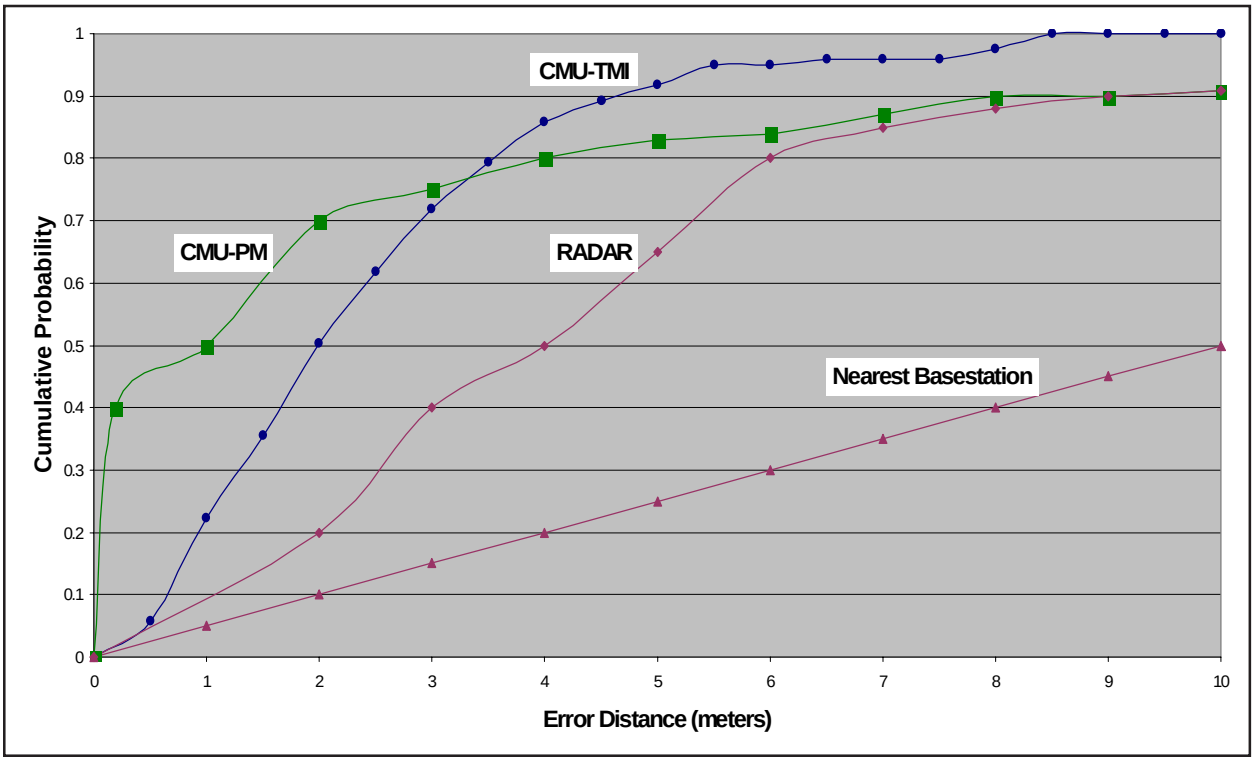

Figure 7 - Cumulative Distribution Function (CDF) Accuracy of Location Sensing Algorithms

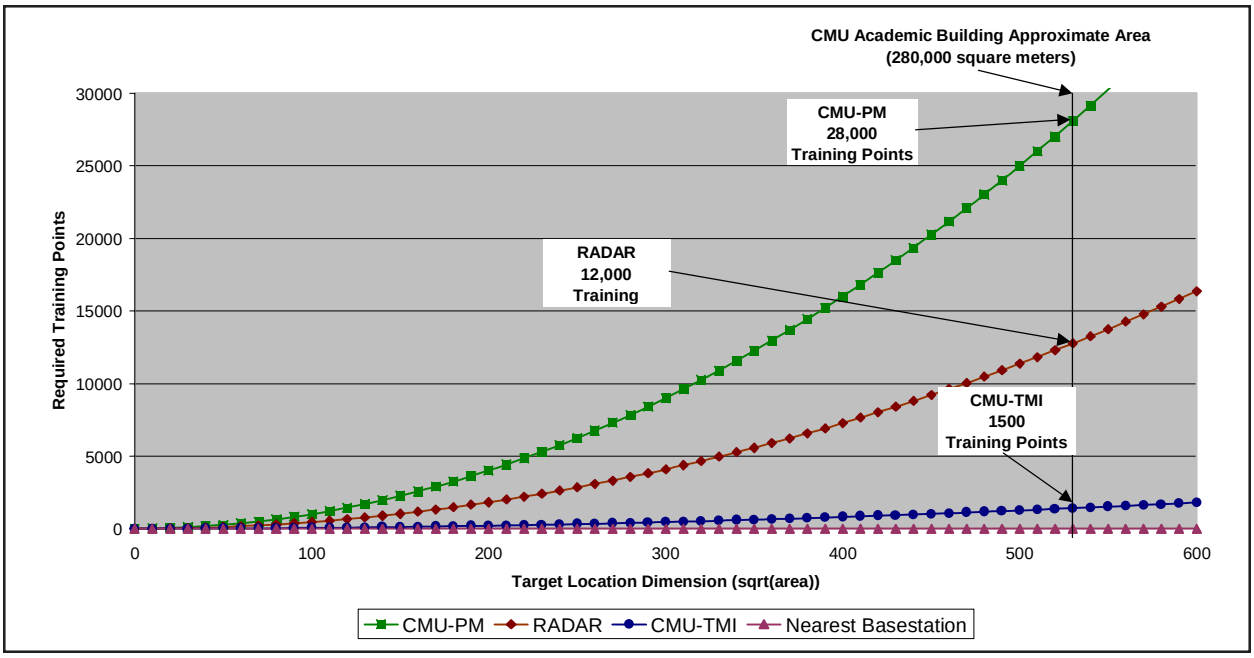

Figure 8 - Training Complexity of Algorithms

puter was evaluated while running the CMU-PM and CMU-TMI location sensing algorithms. The device was fully charged, and run until fully discharged. Location calculations were performed once per second. Figure 9 shows the results of this testing. The CMU-TMI algorithm decreased the battery life by 60% while the CMU-PM algorithm decreased battery life by 75%. Changing the scanning rate from once per second to once per ten seconds reduces battery life impact to less than 6% for CMU-TMI and less than 8% for CMU-PM. The difference can be accounted

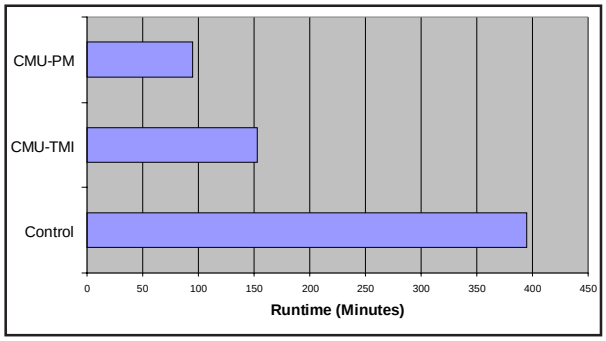

Figure 9 - Power Consumption of CMU algorithms

by the number of signal strength scans, the CMU-TMI algorithm averages 5 scans per calculation, while the CMU-PM algorithm averages 10 scans per calculation.

Each method has its strengths, and weaknesses. The nearest basestation is the simplest algorithm to implement. It requires no field training. However, this algorithm produces very low resolution location results. RADAR and CMU-PM are fundamentally similar and are thus categorized together. The results from these algorithms are as good as their trained data, a higher accuracy requires a larger effort in training. CMU-TMI provides an 8 times reduction in complexity over the RADAR algorithm, and generates more accurate results. The accuracy is not as good as the best pattern matching algorithm, but the efficiency in training provides a distinct advantage for large areas.

6. Privacy of Location Information

Location information enables services and conveniences for the users of such a system. This raw functionality costs the users their privacy. To enable users to maintain control of their location information, research has been done in determining how and when users want to control this information.

6.1. Model and Methodology

The privacy of location information is described by set theory and rules. Each rule establishes a list of users who is allowed or disallowed to know the location of a user for a given duration of time. A rule establishes a time duration and possible repetition of an event. The rule sets authorization based on one of four visibilities, Visible to All, Invisible to Some, Visible to Some, and Invisible to All. These visibilities are arranged as increasing restrictiveness of the set. Visible to All allows anyone to know the location of a

client user, Invisible to Some restricts only a finite list of users, Visible to Some restricts all users excepts a finite number of users, and Invisible to All restricts everyone.

If these user's rules are combined with a boolean AND, two conflicting rules will result in the location information not being available to the user with the conflict. Likewise, by combining rules with a boolean OR, two conflicting rules will result in the location information being available to the user with the conflict. Figure 10 illustrates this effect of using AND or OR.

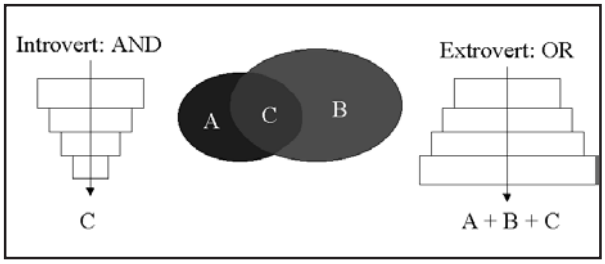

Figure 10 - Privacy: Introvert and Extrovert Behavior

6.2. User Studies and Results

The user study consisted of a series of questions related to how a user chooses to control personal information. The study included 53 CMU undergraduate and graduate students and 1 CMU staff member. Questions related to instant messaging, scheduling habits, and direct location information handling were included. The first question, "Who has authorization to send you a message?" can be answered "Anyone" meaning there is no authorization required, "Interactive" meaning each user must be separately authorized with an interactive mechanism, and "Setup" meaning each user must have prior authorization. Figure 11 presents the results of this question. This question relates to how users give access to themselves, yet maintain control of the transfer of personal information. This question offers a baseline to compare how users give access to the automatic transmission of personal information, such as location. Users appear to be equally likely to choose automatic acceptance or interactive acceptance of authorization and are half as likely to rigorously setup authorization. The second question "Who can see my online status" can be answered in the same way as the first question. Figure 12 shows the results of this question. This question is meant as an analog to the automatic transmission of personal information. This question shows a bimodal distribution where twice as many people appear willing to automatically transmit their personal information to any user, and a second group choosing greater privacy by rigorous setup of who is allowed to inquire of their personal information. Few users choose to interactively transmit their information for every request. The third question relates to how to resolve conflicts between multiple privacy rules. If one rule states user A is authorized to see the client at the current time, and another rule states the opposite, which rule will take precedence. A situation describing this was presented to the users, their responses can be "Cannot See" meaning the client wishes their information not to be given if a conflict exists, and "Can See" meaning the client wishes their information to be given if such a conflict exists. Figure 13 shows the responses from this question. This question is meant to establish the nature in which rules will be resolved. We expect this to be

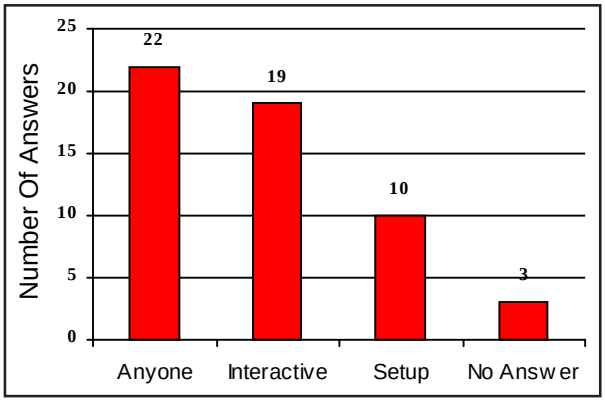

Figure 11 - Privacy: "Who can communicate with you?"

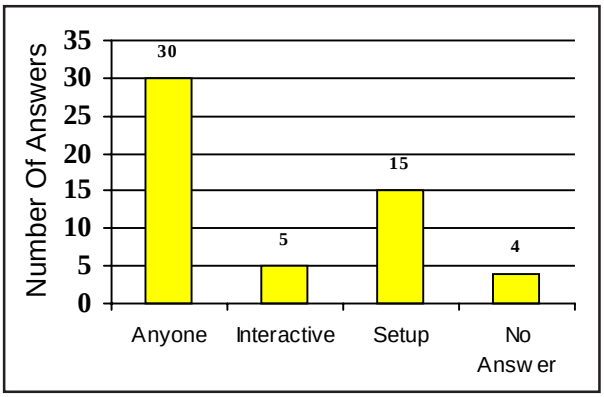

Figure 12 - Privacy: "Who can know your online status?"

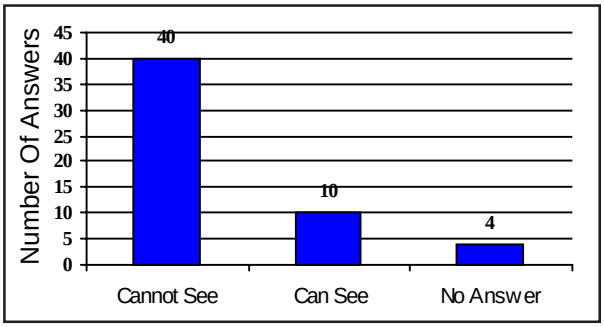

Figure 13 - Privacy: Authorization Conflict Resolution

analogous to determine if a user is an introvert, or an extrovert. The last question, "What is the smallest unit of time used on your schedule?" establishes the granularity with which users choose to regulate their time. This relates to the expected minimum time a cached value of a schedule is valid for. As shown in Figure 14, it appears that for most uses, 10 minutes is an appropriate time for the measure.

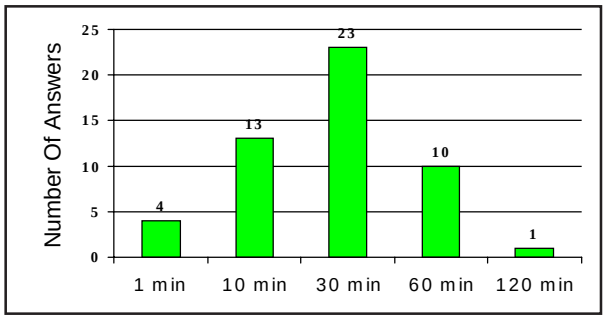

Figure 14 - Privacy: Scheduling Granularity

7. Future Research

While context information is useful for generating more intelligent behavior in systems, this information is a liability for the users of the system. Location information is a prime example. The security of this information must be addressed at all levels of the system, including the architecture, protocols, inferred preferences and user specified preferences.

The location service is not optimized at this time. Requiring the tracked client to return it's current location for every request uses power and compute cycles of the limited mobile devices. Ideas for increasing the efficiency of the location service include caching and predicting the location of users.

More research is needed on how to efficiently and practically generate rules to govern privacy. Current ideas include automated wizards to assist users and the use of machine learning or pattern discovery to automatically generate rules for each user. New privacy rules may set a users visibility based on their present location and inferred task, as well as timed events.

8. Conclusions

This paper presents and evaluates two location sensing algorithms and compares their performances and merits against other existing algorithms. We introduce attributes for characterizing location sensing systems, accuracy, complexity of training system and power consumption. Our results provide guidelines under what circumstances one location sensing algorithm may be used in preference to another. Our method reduces training complexity by a factor of eight, and yields noticeable better accuracy. We have designed and implemented a privacy model and performed user studies. Users expect two unique behaviors from the system, an introvert model where privacy is preferred, and an extrovert model where availability of information is preferred.

9. Acknowledgments

We would like to acknowledge the support provided by IBM Research, Hewlett Packard, Lucent, the National Science Foundation, Pennsylvania Infrastructure Technology Alliance and the Defense Advanced Research Projects Agency.

10. References

[1] Bahl, P., Padmanabhan,V. N., "RADAR: An In-Building RF-Based User Location and Tracking System," Proc. IEEE Infocom 2000, Tel Aviv, Israel, March 2000.

[2] Bahl, P., Balachandran, A., Padmanabhan,V. N., "Enhancements to the RADAR User Location and Tracking System," Microsoft Research Technical Report, February 2000.

[3] Clarkson, B., Pentland A., "Recognizing User Context via Wearable Sensors." Proceedings of the Fourth International Symposium on Wearable Computers (ISWC'00), Atlanta, GA, 2000.

[4] Rungsarityotin, W., Starner, T. E., "Finding Location Using Omnidirectional Video on a Wearable Computing Platform." Proceedings of the Fourth International Symposium on Wearable Computers (ISWC'00), Atlanta, GA, 2000.

[5] Smailagic, A., Siewiorek, D.P., "User-Centered Interdisciplinary Design of Wearable Computers", ACM Mobile Computing and Communications Review, Vol.3, No.3, pp43-52, 1999.

[6] Small, Jason, "Location Determination in a Wireless LAN Infrastructure", Masters thesis, Department of Electrical and Computer Engineering, CMU, 2000.

[7] http://www.igeb.gov/

[8] http://www.uk.research.att.com/spirit/