Dynamic Optimization of Bluetooth Networks for Indoor Localization

Markus Jevring
Telematica Instituut
The Netherlands
markus@jevring.net

Robert de Groote
Telematica Instituut
The Netherlands
robert.degroote@telin.nl

Cristian Hesselman
Telematica Instituut
The Netherlands
cristian.hesselman@telin.nl

ABSTRACT
Ubiquitous computing environments typically contain a large number and a large variety of networked sensors that are often embedded in the environment. As these networks grow in size and complexity, their management becomes increasingly costly, for instance in terms of equipment, software, and people. One way to keep these costs under control is to automate some or all of the management aspects in the system, reducing or even removing the need for human interaction. In this paper, we focus on automatically managing Bluetooth networks for indoor localization, which we consider a specific class of ubiquitous computing systems because they usually rely on many Bluetooth devices scattered throughout a particular building. We will discuss algorithms that help reducing the number of active devices needed in a network, while maintaining a comparable localization accuracy compared to the "full" network. The algorithms enable the most "valuable" Bluetooth devices in the network and will disable the others. The main advantage is that this reduces the need for network planning, which reduces the costs of operating the system. Another advantage is that it reduces the amount of energy used by the network and the mobile devices being located. We evaluate the real-world performance of our algorithms through experiments carried out with a running system in a realistic environment. We found that our algorithms can reduce a network to approximately half the original size while still retaining an accuracy level comparable to the original "full" network.

Categories and Subject Descriptors
C.2.4 [**Distributed Systems**]: Distributed applications

General Terms
Algorithms, Management, Performance, Experimentation

Keywords
Ubiquitous computing, sensor networks, Bluetooth, Indoor Localization, Automatic Management, Experimental results

1. INTRODUCTION
The ubiquitous computing paradigm [3] relies on an infrastructure of many networked sensors and devices that allow services to obtain the data that they need for their specific purpose. Like any other computer system, these sensor networks need to be managed. For example, they need to be installed, (re-)configured, and possibly taken out of service. Traditionally such management has been carried out by humans, but as the sizes of ubiquitous computing networks grow, coupled with the fact that the sensors get smaller and smaller, and can be deployed in comparatively inaccessible or even hazardous locations, means that the cost in effort and resources for humans to manage such system can become preventatively large. An automatic management system that takes over some or all management tasks from humans is therefore a crucial component of any realistic ubiquitous computing environment. The extreme of this is that humans will not be needed at all for the day-to-day or even emergency handling of a deployed system.

We have created a system called AMBIENT (Automatic Management of Bluetooth-based Indoor localization NETworks) that helps human operators with the management of Bluetooth networks for indoor location [1]. We think of such networks as a special class of ubiquitous computing systems because they typically rely on many Bluetooth devices scattered throughout a particular building [6]. AMBIENT consists of two main parts: (1) a set of algorithms that determine which Bluetooth sensors should be available and which should not and (2) a remote control infrastructure that allows the algorithms to control the settings and availability of the different sensors in the network.

In this paper, we will look at two algorithms that are used within AMBIENT, the entropy sort and worst contributor removal algorithms. Both algorithms estimate the usefulness of each sensor to the network and aim for reducing the number of active Bluetooth sensors in the network while maintaining a localization accuracy comparable to that of the "full" network (i.e., when all devices are turned on). The entropy sort optimizer uses entropy to estimate the usefulness of each sensor and the worst contributor removal uses the estimated loss in localization accuracy that the removal that would result from the removal of each of the devices from the network. We will discuss how these algorithms work, and we will evaluate their performance by means of real-world experiments conducted with an implementation of AMBIENT in a realistic setting. The contributions of our work are (1) the application of the optimization algorithms to the problem of automatically managing Bluetooth networks for indoor localization and (2) the evaluation of the performance of these algorithms based on real-world experiments.

We begin our discussion with related work in Section 2. In Section 3 we briefly consider the operation of the existing localization system that AMBIENT builds upon, which relies on so-called “fingerprints” to locate Bluetooth devices. Section 4 discusses the optimization algorithms and Section 5 discusses the various performance aspects of the algorithms. Section 6 contains conclusions and future work.

2. RELATED WORK

Our aim is to create sensor networks that contain fewer devices than the full networks. To do this, we must find a way to select only the devices that will best serve our needs, and include them in the network. There are a variety of ways of doing this. We will examine some of these methods in this section.

Reducing the number of devices used in a localization system can then be considered a dimensionality reduction problem [11]. Dimensionality reduction problems are solved by analyzing the values of each variable in a set of variables and retain those that say the most about the data set. Each data set contains a number of vectors that have a certain dimension. We can apply this to our problem by mapping fingerprints to vectors, where the dimensions of the vector equal the number of devices in the fingerprints. Therefore, finding a way to reduce the number of dimensions in a vector translates to reducing the number of devices in a fingerprint, and by extension, reducing the number of dimensions in all the vectors in the data-set lets us reduce the number of devices in the whole network.

One dimensionality reduction algorithm is Principal Component Analysis [9] (PCA). PCA will create a new coordinate system based on an existing set of data. Each axis in the new coordinate system will be created along the most valuable dimension, after the last one created. For instance, imagine a set of data points in 3D shaped like a loaf of bread. The first axis would be the length of the loaf (because it has the widest range of data points, the highest variance). The second axis would be the width of the loaf (the second highest variance), and so on. This can be applied from any number of dimensions to any other number of dimensions, creating new axes along the dimensions that have the highest variance. PCA is not suitable for the problem at hand as we cannot change the possible dimensions of the coordinate system, only chose which dimensions to keep.

Another way to solve a related set of problems is by using decision trees [12]. Decision trees classify objects based on their attributes. The root node of a (sub)-tree is chosen based on the decisiveness of an attribute to classify the object. One such algorithm is ID3 [10]. Created by Ross Quinlan, the Iterative Dichotomizer (ID3) works by calculating the entropy for each of the attribute, and selecting the one with the best entropy to form the node of the current (sub)-tree. This works in a recursive manner until the attributes have been exhausted. This creates a tree where the most valuable attribute is at the top. This property would allow us to select the root node of a tree in an iterative manner, or select a series of root nodes in a pre-order traversal manner to select the set of most valuable devices. Unfortunately, ID3 requires that the variables have discreet values, so that branches can be formed. This requirement also makes it unsuitable for our use, as our fingerprints deal with continuous values.

Yet another way is to use graph-based methods. Cărbunar *et al.* [7] suggests using Voronoi diagrams to reduce the number of sensors required in a network to improve energy efficiency. They use Voronoi diagrams to determine if the sensing area of one sensor is completely covered by a set of other sensors. If so, then that device can be removed. They present both centralized and distributed ways of achieving this. They assume that sensors are unimpeded by objects in the environment, and perform their calculations with perfect circular coverage areas for each sensor. This might be a problem for indoor systems, as radio propagation due to walls and furniture and people do not lead to perfectly circular (or spherical in 3D) coverage areas [8].

One of the algorithms described in this work is loosely based on ID3 in that it uses entropy as a measure to determine which dimensions most uniquely identify the data. The entropy is calculated based on the *a posteriori* probabilities of being in a room, as indicated by each device. The other algorithm does not use entropy, but rather looks at the loss in estimated prediction accuracy as an indicator for which dimensions to keep.

3. LOCALIZATION SYSTEM

AMBIENT builds upon an existing indoor localization system that uses *reference* fingerprints to locate users carrying Bluetooth-enabled devices. A reference fingerprint is a collection of signal characteristics that identify a certain location. The reference fingerprints we use are based on the (inquiry) response rates [6] of the other sensors in the system. A reference fingerprint for a given location consists of a visibility measure (the response rate) for each device that is visible from that location. The (inquiry) response rate is defined as the number of replies received relative to the number of inquiries sent. A device that is part of a fingerprint is said to contribute to that fingerprint. Figure 1 shows an example of a location’s reference fingerprint that has five contributions (device d1 through d5).

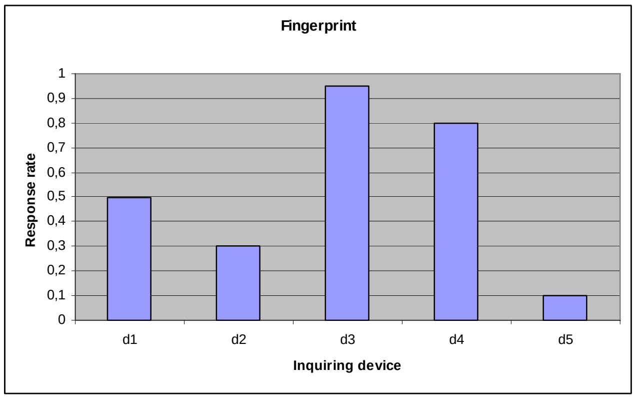

Figure 1. Sample reference fingerprint for a location.

Fingerprints are used by the localization system when determining the location of a device. The system creates an *observed fingerprint* for the device whose location it wants to determine and then compares that *observed fingerprint* to the reference fingerprints of the known locations in the system. Based on these comparisons, each location is assigned a probability stating the likelihood that the device is at that particular location. In other word, the localization system calculates a *location probability distribution*. The location associated with the reference fingerprint that provides the best match for the observed fingerprint has the highest probability and is therefore the most likely location of the device [6]. When running with an optimized network, the reference fingerprints are modified so as to not include devices that are currently not included in the network.

4. ALGORITHMS

We developed two optimization algorithms to deal with the problem of reducing the number of Bluetooth sensors in a network used for indoor localization. These are the *entropy sort optimizer* and the *worst contributor optimizer*. Both of these algorithms optimize for the number of active Bluetooth sensors, targeting a certain localization accuracy level. These algorithms both work with the fingerprints that exist in the system to determine which devices are most suitable to be included in the optimized network. We chose these algorithms because they do not require any other data than those that are already in the system; the fingerprints and the set of available devices.

We believe that these algorithms are applicable to other fingerprint-based systems, if certain conditions are met. These conditions are that the system must use fingerprint comparisons to carry out its work, the system must allow the nodes to be controllable (be able to be turned on and off), and the nodes must be dedicated to the system. If the nodes are not dedicated to the system, then turning them on or off may have adverse effects on any other systems in which they participate.

The following sections contain a brief discussion on the inner workings of these two optimization algorithms and the measure they use for optimizing for certain target accuracy. Alternative optimizers that we used but that we will not consider here, as well as a more detailed discussion of the entropy sort and worst contributor optimizers, are available in [1].

4.1 Accuracy Metric

Our accuracy metric aims to estimate the accuracy of the localization system given a particular constellation of the Bluetooth network. We define the metric as the average chance that a correct location is mistaken for another, incorrect location. For instance, a network for which the estimated accuracy is 0.95 would be expected to be able to accurately determine the location of a given device in 95% of the cases. When the optimizers are executed, they are given a target accuracy to aim for. The accuracy metric is then used to determine if the current network that an optimizer has created has an estimated accuracy that is better than or equal to the target accuracy provided. If it is, then the optimizer returns the network. If not, the optimizer keeps working until it creates a network whose estimated accuracy is better than or equal to the target accuracy.

The accuracy metric used is based on fingerprint clustering, taking normalized Euclidean distance as an inter-fingerprint distance measure. Around each reference fingerprint a cluster is defined which includes all fingerprints that lie at a distance smaller than a certain threshold known as the clustering radius, *R*. Given the number of fingerprints that are part of the cluster around a target location's fingerprint, an error rate for that location can be derived: Assuming the localization system is unable to determine which of the fingerprints in a cluster best matches an observed fingerprint, the system is forced to make a uniformly distributed guess. As a result, the probability of choosing the correct location equals 1 / n, where *n* is the number of fingerprints in the same cluster as the given fingerprint.

The accuracy of the entire network is calculated by averaging the probabilities of correctly choosing each location from the cluster centered around that location's fingerprints. A more detailed description of this is available in [1].

Different values for the clustering radius, *R*, were experimentally evaluated [1]. The value that performed well in these experiments and which is therefore used to obtain the results described in this paper is 0.15.

4.2 Entropy Sort Optimizer

The entropy sort optimizer determines which the most valuable devices are, and adds them to the network in order. This value, and subsequently this order, is based on the entropy [2] *H* of the location probability distribution, and can be formally defined as:

$$H(Loc) = -\sum_{l \in Locations} P(Loc = l) \cdot \log P(Loc = l)$$

We are interested in devices with a low entropy, because these devices will more distinctly indicate each location that they contribute to. A device with a maximum entropy would contribute equally to all fingerprints that it contributes to, and therefore be valueless.

The entropy sort algorithm works as follows; we start with an empty *current* network, and a set of *remaining devices*. We iterate over the set of remaining devices and, for each device in this set; we calculate the entropy of the location probability distribution given the devices already in the current network. The device that yields the network with the lowest entropy is moved from the set of remaining devices to the current network. Each time a device is moved to the current network, we calculate the estimated accuracy of the network, using the metric described in Section 4.1. If the accuracy did not increase after the addition of this latest device, the device is discarded and the algorithm continues with the next remaining device. If the accuracy *did* increase, the accuracy of the current network is compared to the target accuracy, and if the accuracy of the current network is larger than or equal to the target accuracy, the optimizer returns the current network.

4.3 Worst Contributor Removal Optimizer

The worst contributor removal optimizer starts with a full network and continues by removing devices from it until the target accuracy is reached. It determines which device to remove by calculating which device is the worst contributor in the current network. It does this by calculating the decrease in network accuracy the removal of each device would yield. The device that yields the lowest decrease in accuracy is the worst contributor, and as such is removed. The optimizer does this iteratively until the accuracy of the network drops below the target accuracy. When this happens, the last device to be removed is re-added to the network, and that network is returned from the optimizer.

The worst contributor removal optimizer is a brute-force optimizer that was chosen because of its simplicity. It relies on the accuracy calculation discussed in Section 4.1 to do the heavy work. It offers a very intuitive view of relative device values; much more so than the entropy sort optimizer.

5. EVALUATION

To evaluate the performance of our algorithms, we conducted a set of experiments with AMBIENT configuring the localization system of Section 3. This section details the experimental setup, as well as our findings. The detailed results are available in [1]

5.1 Experimental Setup

To conduct our experiments, we set up a controlled environment of 12 fingerprinted locations (reference fingerprints), each with a Bluetooth infrastructure device, in a 3x4 grid; four adjacent rooms spanning 3 floors, as schematically shown in Figure 2. The letters in the figure indicate the locations of the devices whose locations we wish to determine, hereinafter referred to as the *unknown* or *mobile devices*.

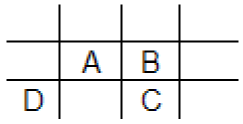

Figure 2. Locations of the four unknown devices used in the experiments

The infrastructure devices are SITECOM Bluetooth USB Adapter CN-500V2 [5] Bluetooth version 1.2, class 2 dongles connected to the desktop PCs of employees in the building. The Bluetooth version used by the mobile devices was also 1.2. We had originally conducted experiments with mobile devices using Bluetooth version 1.1, but this version disparity between infrastructure devices and mobile devices led to poor results [1] due to differences in the inquiry process [14][15].

The experiments were conducted for 20-30 minutes per optimized network. The localization accuracy is based on location determinations done every five seconds during this interval, resulting in 240-360 invocations of the localization system per experiment. From these predictions we get a probability distribution that says how often a certain location was the most likely one according to the system. For example, a prediction accuracy of 75% means that in 75% of these predictions, the localization system predicted the unknown device's correct location to be the most likely one.

The algorithms were executed on a central component in the AMBIENT architecture called a Reconfiguration Decision Point (RDP) [1]. This RDP keeps track of the state of the network, and issues control commands to the nodes participating in the network via the AMBIENT remote management framework.

5.2 Optimized Network Size

Figure 3 shows the sizes of the optimized networks using the Entropy Sort optimizer (ES) and Worst Contributor Removal optimizer (WCR). ES-x and WCR-x represent the algorithms configured for a target accuracy of x%. We calculated Figure 3 in an off-line manner by applying the algorithms to the fingerprinting database of the localization system, which contains a reference fingerprint for each of the fixed Bluetooth sensors in our company's building (more than the 12 we are using in our controlled experiments). The full network of 54 devices was used to create new networks of different sizes. For each size, 10 networks were randomly chosen as the original network and fed into the algorithms using different target accuracies, after which the number of retained devices was measured. The horizontal axis of Figure 3 shows the original size of the network, the vertical axis the average percentage of Bluetooth sensors that the algorithms kept active. Error bars are not included in the graph, but varied from 2% in the largest original networks to 7% in the network consisting of 12 devices.

As can be seen from Figure 3, the worst contributor removal will generally generate smaller networks than entropy sort, given the same target accuracy. It also shows that, regardless of original network size, the reduction ratio is similar for any given optimizer with a particular target accuracy. For instance, the ES-0.95 optimizer delivers a network that contains 55% - 65% of the sensors of the original network, regardless of the size of the original network. These results show that our algorithms satisfy one of our requirements for them, which was that they needed to produce significantly smaller networks than the original network.

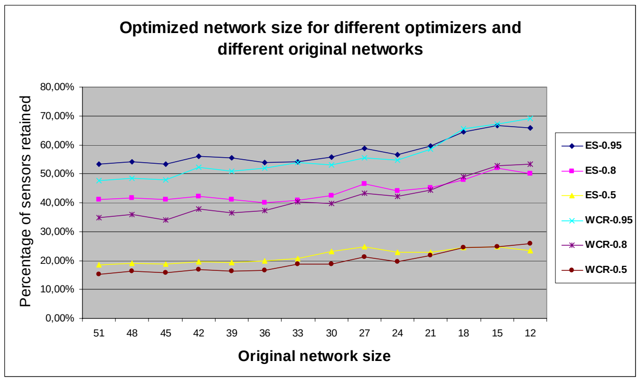

Figure 3. Optimized network sizes for different optimizers and target accuracy levels.

5.3 Optimization Time

Figure 4 visualizes the time taken by the various optimizers to get to the desired accuracy level. The x-axis shows the original network size, as in Figure 3. Figure 4 is also based on the complete contents of the localization system's fingerprinting database.

We can see that, as the size of the original network increases, so does the runtime of the optimizers. We can also see that the runtime increase of the worst contributor removal is drastically higher than that of the entropy sort optimizer. This is due to the fact that the worst contributor removal performs many more accuracy calculations, and these accuracy calculations take more time the larger the network is. As the worst contributor removal starts with a full network and removes devices, the accuracy calculations are more often done on larger networks than on smaller networks. The entropy sort optimizer, however, first does not use the accuracy calculation to select which devices to include, and second primarily performs accuracy calculations on smaller networks rather than larger ones. These are the contributing factors to the results seen in Figure 4. The higher runtime or the worst contributor removal optimizer makes it unsuitable for all but the very smallest networks. Due to this, the remainder of the analysis will focus solely on the entropy sort optimizer.

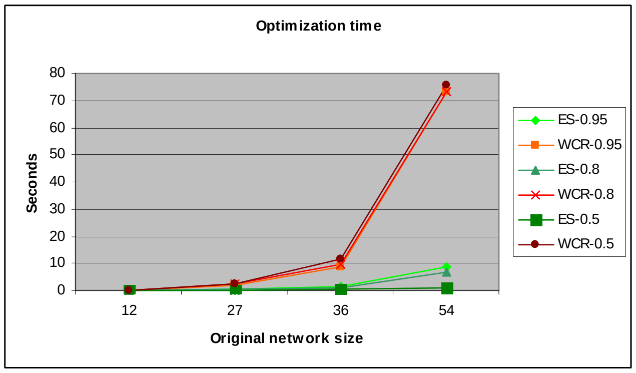

Figure 4. The optimization time for different optimizers and target accuracies given different original network sizes.

5.4 Estimated vs. Real Accuracy

Figure 5 displays the average and standard deviation of the prediction accuracy of different networks that have been optimized by the entropy sort optimizer with different target accuracies. It shows that a higher target accuracy provided to the optimizers results in a higher prediction accuracy of the network when it is actually being used, which suggests that our accuracy metric (Section 4.1) indeed forms a measure of the accuracy of a network.

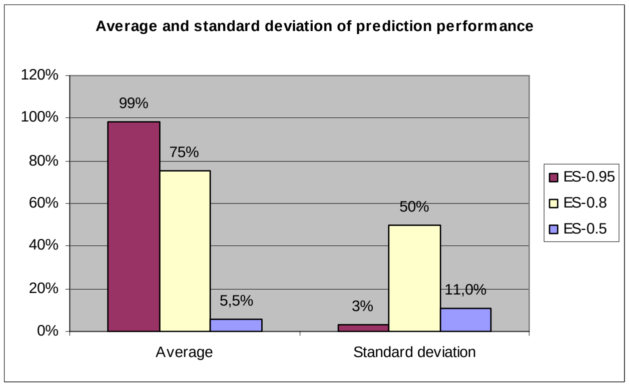

Figure 5. Average and standard deviation of the prediction accuracy (actual accuracy of the localization system) of the entropy sort optimizer using different target accuracies.

5.5 Fingerprint Age

As the age of the fingerprints used in the experiments varied, we discovered that the fingerprint age had a significant impact on the accuracy of the location predictions. Figure 6 shows that there is significant accuracy loss when using fingerprints that are three days old. What it also shows, however, is that younger fingerprints are not guaranteed to be better, as can be seen by the fact that two days old fingerprints perform better than one day old fingerprints. This leads us to believe that what really influences prediction accuracy is the real-time radio frequencies at the time the fingerprints were created compared to the time when they are used. Ideally, we want these conditions to be as similar as possible. One way of achieving this similarity is to create fingerprints as often as possible. It might be possible to correlate the prediction accuracy to some external factor, but that is outside the scope of this paper.

The data that Figure 6 is based on are the averages of the prediction accuracy of the four locations described in Section 5.1. This data is based on three different experiments, each with different fingerprint ages.

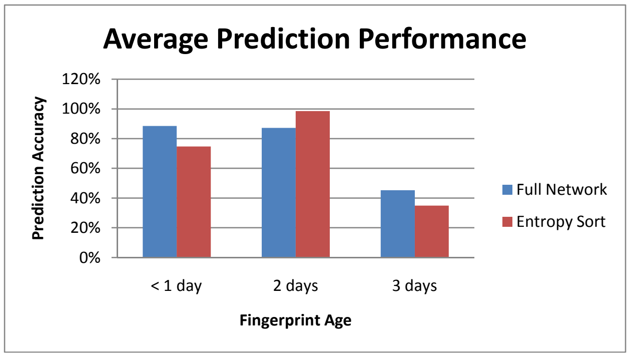

Figure 6. Average prediction accuracy for different networks based on fingerprint age.

6. CONCLUSIONS AND FUTURE WORK

Our results show that it is possible to create an algorithm that can reduce the number of sensors needed in a Bluetooth-based network for indoor location determination while maintaining a comparable accuracy to the full network. We created two algorithms to solve this problem, the entropy sort optimizer and the worst contributor removal optimizer. Our results show that both these optimizers will create significantly smaller networks than the original, retaining 55% - 65% of the devices given a target accuracy of 0.95. Our results also show that the time taken for these optimizers to converge is rather different. The entropy sort optimizer shows signs of acceptable scalability, whereas the worst contributor removal takes too long to execute to be usable in any but the smallest networks.

We also examined the relation between the target accuracy given to the optimizers and the actual prediction accuracy of the resulting network. We found that a network created by an optimizer using a higher target accuracy will have a higher accuracy than a network created by an optimizer using a lower target accuracy, which suggests that our accuracy metric indeed provides an estimation of the accuracy of a network.

Our results also show that fingerprint age has an impact on prediction accuracy. However, since our results do not show a linear relationship between fingerprint age and prediction accuracy, we must conclude that there are some other factors involved. We believe that these factors are the real-world radio conditions at the time the fingerprints are taken and the time they are used. A simple way of reducing the difference in these conditions, whatever they may be, between fingerprint creation and usage is to create fingerprints often, so that there is less likelihood of change in these conditions between creation and usage.

As the creation of fingerprints takes considerable time when the network is large, it becomes difficult to maintain fingerprint freshness at an acceptable level. Further study is needed to determine how we can mitigate the problems caused by these differences, or how to increase the efficiency in the fingerprint creation process, so that it can be done in an acceptable amount of time. One option is to parallelize fingerprint creation process for geographically disparate areas.

7. ACKNOWLEDGMENTS

This work has been conducted within the project Freeband AWARENESS. Freeband is co-sponsored by the Dutch government under contract BSIK 03025.

The authors would like to thank Niels Snoek, Mortaza Bargh and Martin Wibbels for all their wonderful help.

8. REFERENCES

[1] Markus Jevring, “Automatic Management of Bluetooth Networks for Indoor Location”, Master Thesis, University of Twente, 2008.

[2] Shannon, Claude E.: Prediction and entropy of printed English, *The Bell System Technical Journal*, 30:50-64, 1950.

[3] Mark Weiser. “The Computer for the 21st Century.” *Scientific American*, 265, September 1991

[4] H. van Kranenburg, M. S. Bargh, S. Iacob, A. Peddemors, “A Context Management Framework for Supporting Context-Aware Distributed Applications”, *IEEE Communications Magazine*, August 2006, pp. 67-74

[5] http://www.sitecom.com/drivers_result.php?groupid=&productid=176&version=V2;001, Checked 2008-06-25

[6] Mortaza S. Bargh and Robert de Groote, "Indoor Localization Based on Response Rate of Bluetooth Inquiries", To appear in the proceedings of MELT'08, 2008

[7] B. Cărbunar, A. Grama, J. Vitek, O. Cărbunar, “Redundancy and coverage detection in sensor networks”, *ACM Transactions on Sensor Networks (TOSN)*, Volume 2, Issue 1, Pages: 94 – 128, (February 2006)

[8] V. Otsason, A. Varshavsky, A. LaMarca, and E. de Lara, “Accurate GSM Indoor Localization”, *UbiComp 2005*, LNCS 3660, pp. 141–158, 2005

[9] Pearson, K., "On Lines and Planes of Closest Fit to Systems of Points in Space". *Philosophical Magazine* 2: 559–572, 1901

[10] Mitchell, Tom M. Machine Learning. McGraw-Hill, pp. 52-81, 1997

[11] Imola K. Fodor, "A survey of dimension reduction techniques", Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, June 2002

[12] Andrew W. Moore, "Tutorial: Decision Trees", http://www.autonlab.org/tutorials/dtree.html, Checked 2008-07-21

[13] A. Dey, D. Salber, and G. Abowd, “A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications”, *Special issue on context-aware computing in the Human-Computer Interaction (HCI) Journal*, Volume 16 (2-4), 2001, pp. 97-166

[14] Peterson, Baldwin & Raines, ”Bluetooth Discovery Time with Multiple Inquirers”, *Proceedings of the 39th Hawaii International Conference on System Sciences*, 2006

[15] Peterson, Baldwin, Kharoufeh, ”Bluetooth Inquiry Time Characterization and Selection”, *IEEE Transactions on Mobile Computing*, Vol. 5, No. 9, September 2006