

Performance Comparison of Indoor Positioning Techniques based on Location Fingerprinting in Wireless Networks

Tsung-Nan Lin, Po-Chiang Lin
 Graduate Institute of Communication Engineering
 National Taiwan University, Taipei, Taiwan
 Email: {tsungnan, r92942075} @ ntu.edu.tw

Abstract

Appropriate and correct indoor positioning in wireless networks could provide interesting services and applications in many domains. There are Time of Arrival (TOA), Time Difference of Arrival (TDOA), Angle of Arrival (AOA), and location fingerprinting schemes that can be used for positioning. We focus on location fingerprinting in this paper since it is more applicable to complex indoor environments than other schemes. Location fingerprinting uses received signal strength to estimate locations of mobile nodes or users. Probabilistic method, k-nearest-neighbor, and neural networks are previously proposed positioning techniques based on location fingerprinting. However, most of these previous works only concentrate on accuracy, which means the average distance error. Actually, it is not enough to measure the performance of a positioning technique by the accuracy only. A comprehensive performance comparison is also critical and helpful in order to choose the most fitting algorithm in real environments. In this paper, we compare comprehensively various performance metrics including accuracy, precision, complexity, robustness, and scalability. Through our analysis and experiment results, k-nearest-neighbor reports the best overall performance for the indoor positioning purpose.

1. Introduction

The popularity of wireless access infrastructure and mobile devices fulfils people's desire to access the multimedia services ubiquitously. Meanwhile, providing additional context-aware services based on the existing wireless access infrastructure gets more and more interest. Indoor positioning is one of the important techniques to make context-aware services feasible. Many domains get benefits from indoor location information of mobile units to provide useful applications and services, such as museum tour-guide and location-based handoff [1] [2].

Many positioning techniques have been proposed, as introduced in [12-14], and they could be classified into

the following categories, including Time of Arrival (TOA) [23], Time Difference of Arrival (TDOA) [3] [4], Angle of Arrival (AOA) [5], and location fingerprinting [6] [20-22]. However, not all techniques are suitable for indoor positioning because of the complexity in the indoor environments. TOA, TDOA, and AOA all estimate the location based on the triangulation technique which requires the line-of-sight (LOS) between the transmitter and the receiver [24]. For normal indoor environments, it is difficult to find a LOS between the transmitter and the receiver. Therefore, the time and angle of arrival signal would be affected by the multipath effect, thus the positioning accuracy would be reduced. Moreover, the measurement of time and angle of arrival signal requires special hardware, so the cost for the positioning task would be high. The last category, *location fingerprinting*, which could overcome the drawbacks mentioned above, use the received signal strength (RSS) at the sampling locations to build a "radio map" for the target environment. Location fingerprinting performs well for non-line-of-sight (NLOS) circumstances and LOS environments are not required. Thus it is suitable for indoor positioning. There are two stages for location fingerprinting: off-line stage and on-line stage. During off-line stage, site survey is performed in the target environment to collect the RSS from nearby base stations at sampling locations. Some preprocessing may be needed in this stage. In the on-line stage, the positioning techniques measure the RSS in real-time and calculate the estimated location coordinates based on the knowledge built during the off-line stage.

Probabilistic method, k-nearest-neighbor, and neural networks are three popular machine techniques which can be used in location fingerprinting. Youssef et. al. proposed a joint clustering technique for indoor positioning, which is based on probabilistic method [7]. Bahl et. al. proposed an in-building user location and tracking system – RADAR [8], which adopts k-nearest-neighbor to compare multiple data sets and

This work was supported in part by YULON NISSAN Motor Co under grant 94-S-C04-H and by Taiwan National Science Council under grant 93-2213-E-002-057.

pick the one that best matches the observed signal strength. Neural networks have been shown to be capable of approximating any nonlinear function to certain degree of accuracy [10]. Therefore, Battiti et. al. use neural networks to model the nonlinear mapping between the location information and the RSS [15]. However, there is no comprehensive performance comparison about them so far, so it is hard to say which one is better. Most of the previous works only concentrate on the accuracy, which means the average distance error, but it is not enough. In this paper, we perform comprehensive performance comparison based on five important criteria, including *accuracy*, *precision*, *complexity*, *robustness*, and *scalability*. In addition to the accuracy, we analyze the distribution of distance errors, which is called precision. The positioning technique with precision concentrated on small values of distance errors would be preferred, since it is more likely to get small distance error. Complexity considers the computing time and consumed power in the on-line stage, which is critical when we implement positioning techniques in mobile devices that have limited CPU power and battery life. Furthermore, a positioning technique with high robustness could function normally even when some of the observed signal strength is never seen before, or when some base stations are unavailable. Scalability ensures the regular positioning algorithm is still functional when the scope of the environment gets large.

This paper is organized as follows. Section 2 discusses the three indoor positioning techniques based on location fingerprinting. In section 3, we explain the experiment environment and tools using for this paper. Section 4 presents the five important performance criteria and comparison results. Finally, section 5 concludes the paper.

2. Indoor Positioning Techniques based on Location Fingerprinting

Location fingerprinting regards the positioning problem as a pattern classification problem [9]. The estimated location coordinate is calculated based on the radio map built in the off-line stage and the different supervised learning methods mentioned in the following subsections.

2.1 Probabilistic Method

Suppose that there are n location candidates $\omega_1, \omega_2, \omega_3, \dots, \omega_n$. This means there are n classes, and the most appropriate one is chosen based on the posteriori probability. During the off-line stage, the RSS from nearby base stations is measured in these n sampling

locations. Suppose vector \bar{S} is the observed RSS during on-line stage, and then the location candidate ω_i is chosen if its posteriori probability is the highest. The decision rule is:

$$\text{Decide } \omega_i \text{ if } P(\omega_i | \bar{S}) > P(\omega_j | \bar{S}), \\ \text{for } i, j = 1, 2, 3, \dots, n, j \neq i$$

From *Bayes' theorem*:

$$P(\omega_i | \bar{S}) = \frac{P(\bar{S} | \omega_i) P(\omega_i)}{P(\bar{S})} \quad (1)$$

posteriori probability $P(\omega_i | \bar{S})$ is the combination of likelihood $P(\bar{S} | \omega_i)$, prior probability $P(\omega_i)$, and observed evidence $P(\bar{S})$. Since $P(\bar{S})$ keeps the same for one positioning process, and the prior probability $P(\omega_i)$ that a mobile node locates at a specific location is assumed to be the same for all over the target environment, the comparison of the posteriori probability could be considered as the comparison of likelihood:

$$\text{Decide } \omega_i \text{ if } P(\bar{S} | \omega_i) > P(\bar{S} | \omega_j), \\ \text{for } i, j = 1, 2, 3, \dots, n, j \neq i$$

The likelihood of each location candidate is assumed to be the Gaussian distribution. Therefore, by parametric learning, the means and standard deviations of each location candidate could be calculated from the sample data. The base stations in the environment are assumed to be independent, so the overall likelihood of one location candidate can be calculated by directly multiplying the likelihoods of all base stations:

$$P(\bar{S} | \omega_i) = P(S_1 | \omega_i) \times P(S_2 | \omega_i) \times \dots \times P(S_m | \omega_i) \quad (2)$$

where m is the number of base stations and S_j means the RSS from the j -th base station.

In order to interpolate the position coordinates to give more accurate results, we calculate the estimated location (\hat{x}, \hat{y}) as the average of the coordinates of all location candidates by adopting their posteriori probabilities as weights.

$$(\hat{x}, \hat{y}) = \sum_{i=1}^n (P(\omega_i | \bar{S}) \cdot (x_{\omega_i}, y_{\omega_i})) \quad (3)$$

2.2 K-Nearest-Neighbor

K-nearest-neighbor [9] calculates the distances between the observed signal strength and all the sample data sets in the previously-installed database. Assume s_{ij} is the j -th sampling data from the i -th base station, and S_i is the observed signal strength from the i -th base station in the on-line stage, where $i = 1, 2, \dots, m, j = 1, 2, \dots, n, m$ is the number of base stations in the environment, and n is the number of sample data sets. The distance between S_i and the s_{ij} data set for all base stations is:

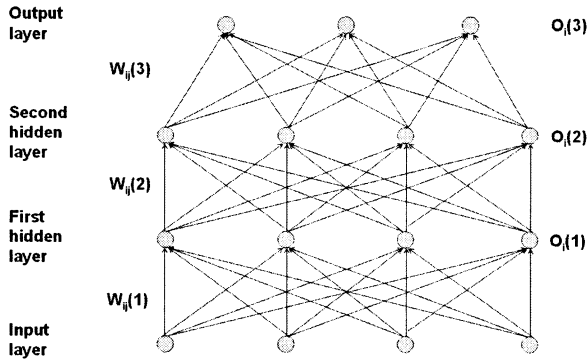


Figure 1. A multilayer perceptron.

$$d_j = \sqrt{\sum_{i=1}^m (S_i - s_{ij})^2}, j = 1, 2, \dots, n \quad (4)$$

After calculated the distance with respect to the whole sample set, a set of k data samples from the database is chosen according the smallest distance. Then, by averaging these k location coordinates, the location (\hat{x}, \hat{y}) of the on-line collected signal can be estimated.

$$(\hat{x}, \hat{y}) = \frac{1}{k} \sum_{i=1}^k (x_i, y_i) \quad (5)$$

where (x_i, y_i) is the location coordinates of the i -th candidate.

2.3 Neural Networks

Neural networks are proved to be effectively constructing the non-linear input-output mapping and have been used in several applications such as classification, approximation, and regularization [10]. The multilayer perceptron (MLP) is one of the most common neural networks that adopt the supervised learning algorithms. MLP consists of a set of sensory units that constitutes the input layer, one or more hidden layers of computational neurons, and an output layer. The input signal propagates through the network in a forward direction, on a layer-by-layer basis. Figure 1 displays a MLP with two hidden layers. The output of the i th neuron at the l th layer can be described as:

$$a_i(l) = \sum_{j=1}^{N_{l-1}} \omega_{ij}(l) o_j(l-1) + \theta_i(l) \quad (6)$$

$$o_i(l) = f(a_i(l)) \quad (7)$$

where $a_i(l)$ and $o_i(l)$ are the activation and output value of the neuron i in the l th hidden layer. The activation is the weighted sum of output from neurons at $(l-1)$ th layer plus the bias term. $\omega_{ij}(l)$ refers to the weight connecting the output from the j th neuron at $l-1$ layer to the input of the i th neuron at the l th layer.

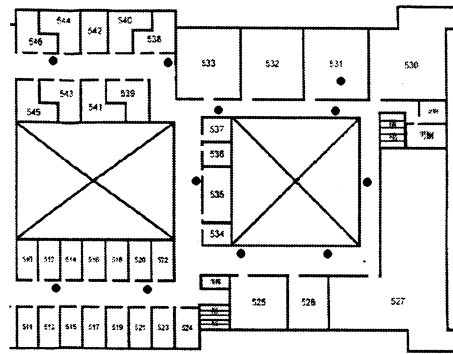


Figure 2. Part of the fifth floor plane of NTU EE, where we perform the experiment. The black dots show the locations of the access points.

The function f is a smooth nonlinear function (i.e., differentiable everywhere). Usually it is a sigmoid function:

$$f(x) = \frac{1}{1 + e^{-x}} \quad (8)$$

or the hyperbolic tangent function:

$$f(x) = \tanh\left(\frac{x}{2}\right) = \frac{1 - e^{-x}}{1 + e^{-x}} \quad (9)$$

The MLP can be viewed as a nonlinear input-output mapping from an n -dimensional Euclidean input space to an m -dimensional Euclidean output space. The universal approximation theorem provides the mathematical justification for the approximation of an arbitrary continuous function [16-19].

3. Experiment Environments and Tools

In order to compare the performance of the three techniques mentioned above, we collect signal strength in a real environment. Figure 2 shows the environment where we perform the experiment. The dimension of the corridor is 24.6 x 17.6 meters. Every place in this environment is covered by at least five IEEE 802.11b APs. We adopt an IBM ThinkPad T40 laptop as the mobile node, with RedHat 7.1 Linux operating system. A Lucent WaveLan/IEEE Wireless Card with Youssef's driver [11] is installed to gather signal strength from nearby APs.

We collect two groups of signal strength for off-line measuring and on-line testing purpose. First we measure signal strength at 42 sampling locations on the corridor for the first group. These sampling locations are separated by 2 meters. At each location we collect 100 samples of the signal strength. Then we collect another set of signal strength in 84 positions at different time for the second group. These locations are separated by 1 meter. At each location we collect 100 samples of the signal strength.

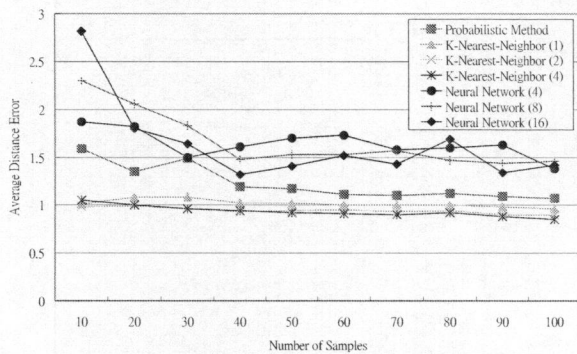


Figure 3. Accuracy of the three positioning techniques

By parametric learning from the sample data, we get the estimate of the means and standard deviations of the likelihoods used in the probabilistic method. In k-nearest-neighbor, there are no preprocessing requirements. We change the number of the nearest neighbors, k , from 1 to 2 and 4 to see the effects. In neural networks, we adopt three-layer MLP topology. Neural Network Toolbox in Matlab is used for designing, training and testing. We adopt sigmoid function as the transfer function of neurons, and set the number of iterations as 1000. We also change the number of hidden layer neurons from 4 to 8 and 16 to see its effects.

4. Performance Criteria and Comparison Results

The following describes the five performance criteria and our experiment results based on the sample data and preprocessing results got in Section 3.

4.1 Accuracy

Accuracy shows how accurate the positioning techniques could achieve. We adopt mean distance error as the performance metric, which is the average Euclidean distance between the estimated location (\hat{x}, \hat{y}) and the true location (x, y) :

$$E(\text{distance error}) = E\left(\sqrt{(x - \hat{x})^2 + (y - \hat{y})^2}\right) \quad (10)$$

Figure 3 shows the average distance errors of the three techniques. In our experiment, the number of samples at each sampling location varies from 10, 20, to 100 with increment of 10.

From the result shown in Figure2, k-nearest-neighbor shows the best accuracy, around 1 meter, among the three techniques. Decreasing the number of samples at each sampling location gets only the minor impact on accuracy. It still gets the accuracy around 1

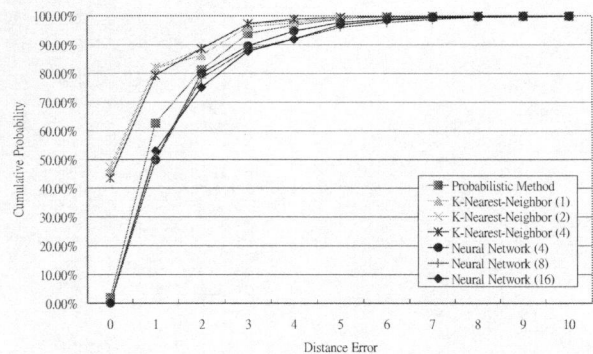


Figure 4. Precision of the three positioning techniques.

meter. Changing the number of nearest neighbor, k , from 1 to 2 and 4, also has no obvious effect on accuracy.

Neural network gets the worst accuracy among the three. However, most of the results show 1.3 to 2.8 meters of distance error, which might be still acceptable in normal indoor environments for most applications. Increasing the number of samples from 10 to 40 would significantly improve the accuracy, while increasing it from 40 to 100 would not. Therefore, 40 could be a sufficient number of training samples for neural networks to efficiently approximately the nonlinear mapping function. Increasing the number of hidden layer neurons would improve the accuracy, but the improvement is not significant.

Probabilistic method gets a middle distance error, about 1.1 to 1.6 meters. Increasing the number of samples at each sampling location would improve its accuracy. This is intuitive because increasing the number of samples would improve the estimation for means and standard deviations of Gaussian distribution.

4.2 Precision

Precision considers the distribution of distance errors, while accuracy only considers the value of mean distance errors. When two positioning techniques are compared, the one with distribution of distance errors concentrated on smaller values would be preferred, since it is more likely that we could get smaller distance error by this positioning technique. We analyze the cumulative probability function (CDF) of distance errors of different positioning techniques, as shown in Figure 4.

From the “precision” point of view, K-nearest-neighbor still shows the best performance. In k-nearest-neighbor, 80 percent of distance errors are within 1 meter. Probabilistic method reports a value of 2

meters to reach the same cumulative probability. In the neural networks, 80 percent of distance errors are within 2 to 2.5 meters, which is dependent on the number of hidden layer neurons. This analysis shows consistent result with the accuracy comparison.

Positioning Techniques	Computing Time
Probabilistic Method	2 seconds
K-Nearest-Neighbor (100 samples at each sampling location)	10 seconds
K-Nearest-Neighbor (10 samples at each sampling location)	1 seconds
Neural Networks	0.25 seconds

Table 1. Computing time of the three positioning techniques.

4.3 Complexity

Complexity considers the computing time required for the on-line stage. When the computation is performed on the mobile node, the impact of complexity would be obvious. Because of the shortage of CPU processing power and battery life in a mobile device, the positioning technique with smaller complexity would be preferred. In the experiments, we repeat each positioning programs for 2500 times on an IBM ThinkPad T40 laptop with Intel 1600 MHz CPU. Table 1 shows the respective computing time.

When applying 100 samples at each sampling location, k-nearest-neighbor requires the most computing time comparing to the other two techniques. Neural network needs the least computing time, but its accuracy is the poorest of the three. There is a tradeoff between complexity and accuracy / precision. It is noted that the complexity of k-nearest-neighbor is strongly dependent on the number of samples at each sampling location, while the other two are not. Decrease the number of samples at each sampling location to 10 would improve the computing time of k-nearest-neighbor to competitive 1 second, while still keeps considerably accurate.

4.4 Robustness

A positioning technique with high robustness could function normally for the following two cases.

Case 1. when some of the observed signal strength is never seen before:

For probabilistic method, since Gaussian distribution is continuous, even the never-seen signal strength would get respective likelihood value.

Therefore, probabilistic method is able to function normally.

For k-nearest-neighbor, the arbitrary value of the signal strength could be used to calculate the distance in the signal space. Therefore, the never-seen signal strength would not be a problem.

Since neural networks have the generalization capability, the never-seen signal strength would not be a problem, either.

Case 2. when some base stations become unavailable:

Assume that the u -th base station is unavailable. Its signal strength is S_u . The number of available base stations changes from m to $(m - 1)$.

For probabilistic method, we simply set the likelihood $P(S_u | \omega_i) = 1$ for all location candidates, and then formula (2) could function as usual.

For k-nearest-neighbor, we change from m -dimensional to $(m - 1)$ -dimensional signal space, and then k-nearest-neighbor functions normally.

For neural networks, we could assign a value smaller than all values we seen before (-100 dBm, for example) to S_u , so the positioning could still function. However, choosing a representative value requires a careful consideration.

4.5 Scalability

Scalability ensures an algorithm can reuse its knowledge (parameters constructed previously) when the scope of the environment gets large. Usually the radio cover range of a base station has some limit. A large area might not be covered by the same set of base stations. Some positioning techniques may require extra preprocessing to handle this problem.

Assuming there are totally M base stations in the whole environment, and we collect data from m base stations during the on-line stage.

For probabilistic method, we first check the database. If the location candidate ω_j is not covered by these m base stations, we set its posteriori probability $P(\omega_j | S)$ to be 0, which means that the mobile node is probably not at this location candidate

For k-nearest-neighbor, only the subsets of sample data that contain these m base stations are chosen. Then we perform calculation as subsection 2.2 describes according to these subsets of sample data.

Neural networks require the input base stations to be fixed, in order to be trained and tested. We have to divide the whole area into several smaller sub-areas. Each sub-area is covered by a set of base stations. We design and train a neural network for each sub-area. While performing positioning, first of all, we have to decide which sub-area the mobile node may belong to by checking the sensed base stations, and then we can

perform positioning by means of the specified neural network using in that sub-area.

5. Conclusion

For many location-based services and applications, it is crucial to find out an appropriate and correct positioning technique. The probabilistic method, k-nearest-neighbor, and neural networks are techniques of estimating the location based on information of received signal strength. In this paper, we achieve comprehensive comparison of three indoor positioning techniques based on location fingerprinting. We compare performance of five performance criteria, including accuracy, precision, complexity, robustness, and scalability. Additionally, we find out that k-nearest-neighbor gets the best performance with high accuracy and precision, and its complexity could be reduced to a competitive level, while still keep good accuracy. Furthermore, k-nearest-neighbor could handle the robustness and scalability issues by little or no extra efforts.

6. References

- [1] S. Thrun, "Probabilistic algorithms in robotics," *AI Magazine*, 21(4):93-109, 2000.
- [2] M. H. Jin, H. K. Wu, and J. T. Horng, "An Intelligent Handoff Scheme Based On Location Prediction Technologies", *IEEE European Wireless 2002*. pp. 551 – 557, Feb. 2002
- [3] M. A. Spirito, "On the Accuracy of Cellular Mobile Station Location Estimation", *IEEE Transactions on Vehicular Technology*, Vol. 50, No. 3, May 2001.
- [4] T. S. Rappaport, J. H. Reed, and D. Woerner, "Position location using wireless communications on highways of the future", *IEEE Comm. Magazine*, Vol. 34, pp. 33–41, Oct. 1996.
- [5] S. Tekinay, "Wireless Geolocation Systems and Services," Special Issue of the *IEEE Comm. Magazine*, April 1998
- [6] M. Brunato and C. K. Kalló, "Transparent Location Fingerprinting for Wireless Services," technical report DIT-02-071, *Informatica e Telecomunicazioni*, University of Trento, 2002
- [7] M. A. Youssef, A. Agrawala, A. U. Shankar, "WLAN Location Determination via Clustering and Probability Distributions," *IEEE International Conference on Pervasive Computing and Communications*, 2003.
- [8] P. Bahl and V. N. Padmanabhan. RADAR: An In-Building RF-based User Location and Tracking System. In *IEEE Infocom 2000*, volume 2, pages 775-784, March 2000.
- [9] R. O. Duda, P. E. Hart, D. G., "Pattern Classification," Second Edition, John Wiley, 2000
- [10] S. Haykin, "Neural Networks – A Comprehensive Foundation," Second Edition, Prentice-Hall, 1999.
- [11] M. A. Youssef's Web Page, <http://www.cs.umd.edu/users/moustafa/>
- [12] J. Hightower, G. Borriello, "Location systems for ubiquitous computing," *IEEE Computer Magazine*, volume 34, issue 8, Aug. 2001, pp. 57-66.
- [13] K. Pahlavan, X. Li, J. Makela, "Indoor geolocation science and technology," *IEEE Communications Magazine*, February 2002.
- [14] Y. Zhao, "Mobile phone location determination and its impact on intelligent transportation systems," *IEEE Transactions on Intelligent Transportation Systems*, Vol. 1, No. 1, March 2000.
- [15] R. Battiti, T. L. Nhat, A. Villani, "Location-aware computing: a neural network model for determining location in wireless LANs," Technical Report DIT-02-0083, Department of Information and Communication Technology, University of Trento, Feb. 2002.
- [16] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of Control, Signals, and Systems*, 2(4):303-314, 1989.
- [17] K. Funahashi, "On the approximate realization of continuous mappings by neural networks," *Neural Networks*, 2(3):183-192, 1989.
- [18] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, 2(5):359-336, 1989.
- [19] B. Irie and S. Miyake, "Capabilities of three-layered perceptrons," in *Proceedings of the IEEE International Conference on Neural Networks*, volume 1, pages 641-648, San Diego, CA, 1988. IEEE.
- [20] K. Kaemarungsi, P. Krishnamurthy, "Properties of indoor received signal strength for WLAN location fingerprinting," in *Proceedings of the First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Service*, 2004.
- [21] A. Taheri, A. Singh, E. Agu, "Location fingerprinting on infrastructure 802.11 wireless local area networks (WLANs) using Locus," in *Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks*, 2004.
- [22] K. Kaemarungsi, P. Krishnamurthy, "Modeling of indoor positioning systems based on location fingerprinting," *IEEE INFOCOM 2004*.
- [23] P. H. Dana, "Global positioning system overview," <http://www.colorado.edu/geography/gcraft/notes/gps/gps.html>
- [24] J. Hightower, G. Borriello, "Location sensing techniques," technical report UW-CSE-01-07-01, Computer Science and Engineering, University of Washington, August 8, 2001.