Indoor Location Tracking System Using Neural Network Based on Bluetooth

Ashraf Mohamed Ali Hassan

Department of Electronics and Communications Engineering,
October University for Modern Sciences and Arts,
6 October City, Egypt,
ashrafmohali@yahoo.com

***Abstract*— This paper introduces the design and implementation of a Bluetooth based on indoor location tracking system. This system utilizes the integrated Bluetooth modules in any today's mobile phones to specify and display the location of the individuals in a certain building. The proposed system aims for location tracking/monitoring and marketing applications for who want to locate individuals carrying mobile phones and advertise products and services. It is an integrated embedded and desktop system that helps the user to get the location of customers/inhabitants/employee within a certain region. The system is composed of a Server Module which is a java application that runs over desktop PC and is used to display the locations of the nearby mobile phones and send location based advertising message. This paper is aimed also to enhance the system positioning estimation accuracy by choosing the suitable number of neurons used in the neural network.**

Keywords— Objects, Customers, Mobile phones, Services, Accuracy, Neurons.

I. Introduction

In a home environment, remote positioning is more useful than local positioning since the user carrying the mobile device would already know their position, whereas the base station computer would not. With this location information, the base station computer can provide services to the user, for example context sensitive medication prompting or location-based elder activity monitoring [1].

This paper introduces the design and implementation of a Bluetooth-based Indoor Location Tracking System (ILTS) that uses the integrated Bluetooth modules in any today's mobile phones to display the location of the individuals in a certain building.

The proposed system consists of a server module which is a java application that runs over desktop PC and is utilized to display the locations of the nearby mobile phones and the second step after that is to send the location-based advertising control message signal. The system also consists of locating modules which can be written in C and run over the Bluetooth-enabled boards which are used to periodically collect the Bluetooth addresses of the nearby devices/mobiles accompanied with their sensed signal strengths. After that it sends them to the Server Module to process, specify, and display the location of the mobile phones.

II. Related Work

Indoor location tracking systems is used to deliver usefulness to end users, if it is designed and maintained appropriately. It is important to note that there is no one perfect location system. Each system must be evaluated based on the intended application across a variety of dimensions such as its accuracy, the infrastructure requirements [2].

A. Globel Positioning System(GPS)

GPS is the one of the most popular outdoor location tracking system worldwide. GPS first designed for military applications, but today, GPS can be considered solutions for many civilian and consumer applications, such as in-car navigation systems and marine navigation. GPS typically does not work well in most indoor settings, because of constant occlusions which come from the GPS satellites [3].

B. Cricket

The Cricket location support system utilizes ultrasound emitters to make the infrastructure and embeds receivers in the object being located. This approach allows the objects to make all their own triangulation computations. Cricket utilizes the radio frequency signal not only for synchronization of the time measurement, but also to delineate the time region during which the receiver should consider the sounds it receives [4].

III. System Description

The proposed system uses the integrated Bluetooth modules in any today's mobile phones to display the location of the individuals in a certain building. The proposed system aims for smart home location tracking/monitoring applications and marketing applications for whom want to locate individuals carrying mobile phones.

The server module processes and displays the location of each mobile phone by means of its Message Authentication Code (MAC) address using table and a simplified graphical map [5]. Moreover the administrator can set certain messages to be sent if the mobile phone is near a certain place, where the server module can send the settled message to the mobile users.

The proposed algorithms based mainly on measurement of the Received Signal Strength Indicator (RSSI) this

978-1-4673-9939-5/16/$31.00 ©2016 IEEE

measurement is used as input to an algorithm that estimates the location of the mobile node that algorithm utilizes the use of neural network rather than specific measurement values and it targets the same order of simplicity as the I^3BM algorithm, but with the high accuracy of the environment adaptive [6,7] algorithm. To do so; it includes a training phase, for modeling the surrounding environment and calculating the internal weights of the neural network, and location determination phase [8]. Room-level location is selected, where rooms refer to physical rooms or specific logical subspaces, as this is the most needed for smart home applications or generally indoor environment [9].

IV. System Architecture

The system architecture is shown in Figure 1

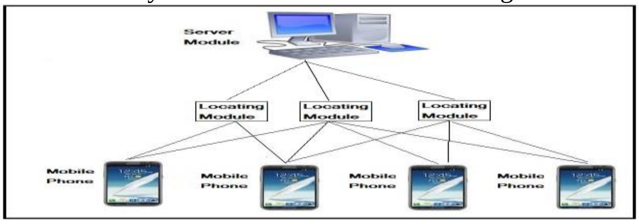

Fig. 1. System Architecture.

A. *Server Module*

It is a JAVA application runs over desktop personal computer and it is used to process, specify and display the locations of the nearby mobile phones and send location-based advertising message [1]. This application utilizes a Java neural network library, by Neuroph, that supports creating, training and saving neural networks.

B. *Locating Modules*

It is a C program that runs over the Bluetooth-enabled Arduino boards. It is used to periodically collect the Bluetooth addresses of the nearby devices accompanied with their signal strengths and send them to the Server Module to process, specify, and display the location of the mobile phones. Each Locating Module covers a number of nearby mobile phones according to their location and received signal strength. In this study there are four Bluetooth-enabled boards used as prove of concept. Figure 2 shows rooms layout.

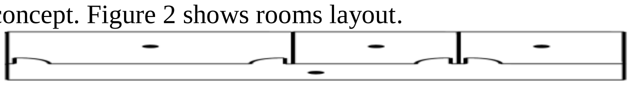

Fig.2. Rooms layout.

C. *Mobile Phones*

Bluetooth service enabled and set to visible. The administrator will just invoke the application and will communicate with it appropriately to explore nearby mobile.

V. Tracking System

Indoor location tracking/monitoring approach based mainly on measurement of signal characteristic as the RSSI, the Link Quality Indicator (LQI), the Time of Arrival (TOA), or the Angle of Arrival (AOA) [7]. This measurement is used as input to an algorithm that estimates the location of the mobile node; in this study we select RSSI as our measure. Measuring RSSI is very simple, it is implemented in all Bluetooth protocol stacks and is available for all Bluetooth platforms. Hence, there are many indoor location monitoring/tracking schemes that utilize this metric as the measurement for location estimation [10]. Among the most popular are the I^3BM [11] and the Environmental Adaptive [2,6,7].

A. *I^3BM*

The I^3BM targets room-level position estimation. It positions the mobile node in the room whose beacon node yields the highest RSSI. To reduce the high sensitivity of the measured values to the surrounding environment, the I^3BM employs mechanisms as signal filtering and room adjacency utilization [12].

As there are many factors influencing the signal readings, it is irrational to base the positioning decision upon a single measurement, but rather upon the outcome of filtering a group of signal values [2,11].

Room adjacency utilization also improves accuracy of location estimation where people move according to feasible paths. Hence, only rooms that are adjacent to current location should be considered as possible estimate for the new location. Figure 3 demonstrates the room adjacency relationship for the layout in Figure 2.

B. *Environmental Adaptive Algorithm*

The Environmental Adaptive scheme has been developed [6,7] driven by the goal of having an accurate position estimation scheme that is robust to environmental changes. The scheme has two phases, calibration phase and localization phase [3]. In the calibration phase, environment characteristics are modeled by computing the m × m transformation matrix, T, which relates the distance between the m anchor nodes and the RSSI of the packets communicated between them using the given equation

$$T = DS^{-1} \qquad (1)$$

Where D denotes the m × m distance matrix whose element d_{ij} represents the known Euclidean distance between the i^{th} and j^{th} anchor nodes, and S denotes the m × m strength matrix whose element S_{ij} represents the RSSI value of the packets between the i^{th} and j^{th} anchor nodes.

In the localization phase, given T and the m-dimensional vector I, whose element I_i represents the RSSI value of the packets between the mobile node and the i^{th} anchor node, then

$$V = TI \qquad (2)$$

Where V denotes the m-dimensional vector whose element V_i is the calculated distance between the mobile node and the i^{th} anchor node. The mobile node position estimate, P(x, y) is then determined using the following gradient descent equation [13,14].

$$P_{K+1} = P_K + \propto \sum_{i=1}^{m} \left(1 - \frac{v_i}{E(A_i, P_K)}\right)(A_i - P_K) \qquad (3)$$

Where P_k is the mobile node position estimate at the k^{th} iteration, A_i is the position of the i^{th} anchor node, α is the gradient step size, and $E(A_i, P_k)$ is the Euclidean distance between the ith anchor node and the mobile node position estimate. At the first iteration, the scheme assumes that the mobile node is directly located at the position of the anchor node with the minimum v_i. The estimated mobile node position, (x, y), is finally mapped to a specific room as per the space dimensions of the underlying environment.

The Environmental Adaptive scheme has demonstrated good performance and much less sensitivity to variations in the surrounding environment [15,16]. However, the inverse in (1) and the iterative nature in (3) are computationally expensive and necessitate slow response and large memory requirements. The Proposed Neural Network scheme is mostly used for fuzzy, difficult problems that don't yield to traditional algorithmic approaches. The proposed system utilizes a neural network which could be described as "4 × 6 × 4 neural network" where it has one input layer with four neurons, one hidden layer with six neurons and one output layer with four output neurons.

C. Developing Environmental Radio Map

The first step is creating a radio map of the area. Measurement locations were chosen roughly every one meter, as shown in Figure 3, excluding the places that were covered by tables or desks. The measurements were taken by a person who was holding the mobile phone, to simulate real usage. Because Bluetooth signals are absorbed by the human body, the direction of the person with regard to the beacon can influence the signal. We therefore measured the signals in four different directions, taking 1 sample in each direction (facing north, east, west and south). Overall measurements were taken on 152 different locations, giving a total of 608 fingerprints.

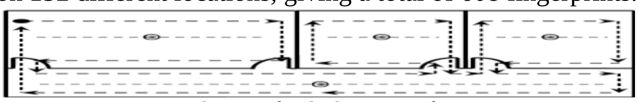

Fig. 3. Radio Map for the locations of measurements

These measurements are applied to a neural network within the JAVA program as a training set. Primarily we utilized a tool named “Neuroph” which is a lightweight Java neural network framework to develop common neural network architectures.

Every neural network has exactly one input layer; and the number of neurons comprising this layer is completely and uniquely determined as the number of columns in your data which is four in our case. Also additional node as a bias node is included, Bias neuron is very important, and the error-back propagation neural network without Bias neuron does not learn. A bias input always has the value of 1. Without a bias, if all inputs are 0, the only output ever possible will be a zero [16, 17].

Like the Input layer, every neural network has exactly one output layer. Determining its size (number of neurons) is simple; it is completely determined by the chosen model configuration. In this experiment the neural network is used as a classifier, and then it has number of output neurons equal to four which is the number of rooms to classify between them.

Finally, concerning the hidden layers since one hidden layer is sufficient for the large majority of problems where there is a consensus that is the performance difference from adding additional hidden layers: the situations in which performance improves with a second (or third, etc.) hidden layer are very small. Now to determine the correct number of neurons to use in the hidden layers several trails are performed that guided by the following rule-of-thumb [15]:

- The number of hidden neurons should be between the size of the input layer and the size of the output layer.
- The number of hidden neurons should be 2/3 the size of the input layer, plus the size of the output layer.
- The number of hidden neurons should be less than twice the size of the input layer.

These three rules provide a starting point for the guided trial and error process, with the help of the visual tool "Neuroph Studio"; we reach to the optimal size of hidden layer to be six neurons. Finally the utilized neural network can be described as “4 × 6 × 4 neural network” where it has one input layer with four neurons, one hidden layer with six neurons and one output layer with four output neurons.

D. Obtaining a Representative Neural Network Procedure

In order to train neural network these data set have to be normalized. Normalization implies that all values from the data set should take values in the range from 0 to 1. For that purpose it would be used the following formula:

$$x_n = \frac{x - x_{min}}{x_{max} - x_{min}} \qquad (4)$$

Where
X is the value that should be normalized
X_n is the normalized value
X_{min} is the minimum value of X
X_{max} is the maximum value of X

The output is represented by 4 digits of data set which represent the target room. For example, “1 0 0 0” represent office room, “0 1 0 0” meeting room, “0 0 1 0” lab room and “0 0 0 1” cafeteria room.

The Neuroph wizard was used to create a Neuroph project, during this wizard there are some settings that we used to create the neural network. Among these setting is the type of training, which is set to supervised training which is the most common way of neural network training[3, 18]. As supervised training proceeds, the neural network is taken through a number of iterations, until the output of the neural network matches the anticipated output, with a reasonably small rate of the error. To achieve the main goal and decide which the best solution for this problem is, several neural networks are created with different settings. Among these settings is the number of neurons in the hidden layer which play a vital role in the whole system performance. Five, six, seven and eight neurons in the hidden layers are tried and the best performance is estimated and presented in the results section.

For testing the neural network, the experimental data were divided randomly into two sets (training and testing dataset) [18]. Separation of 70/30 dataset was done for training/testing where 70% of the experimental data used to train the neural network and 30% of it for testing [19]. Simply Neuroph framework is used to test the performance of the trained network.

This is repeated several times within Neuroph framework till reaching to the optimal results then the corresponding parameters are utilized in the final Java application.

VI. System Configuration

A. Locating Module Setup

Setup one can be used in locating module in each room. It is recommended to have the locating modules separated by minimum of 4 meters and maximum of 10 meters.

B. Arduino BlueTooth Board

The Arduino BT is a microcontroller board based on the ATmega328 and the Bluegiga WT11 Bluetooth module [20,21].

It supports wireless serial communication over Bluetooth (but is not compatible with Bluetooth headsets or other audio devices). It has 14 digital input/output pins (of which 6 can be used as Pulse Width Modulation (PWM) outputs and one can be used to reset the WT11 module), 6 analog inputs, a 16 MHz crystal oscillator, screw terminals for power, an ICSP header, and a reset button.

C. Server Module Setup

A computer device that has a Bluetooth dongle can be connected through the Universal Serial Bus (USB) port[5]. The computer device runs the developed JAVA application, named Server Module, which could connect to all locating modules to sent command and receive response.
The running java application has a suitable User Interface (UI) which allows for users to easily benefits from different capabilities of the system[21].

D. System Hardware Requirements

Table I shows hardware resources

Table I. Hardware resources.

Resources name	Description
Mobile Phone	Any mobile phone that supports Bluetooth and Messaging
Arduino Board	Bluetooth Enabled board
ATMEGA328P Programmer	To burn the Arduino application on the board
Server PC	To deploy/run Server Module
USB Bluetooth Dongle	Attached to the Server PC

E. System Software Requirements

Table II shows software resources

Table II. Software resources

Resources name and version	Description
NetBeans IDE	Used for developing the Server Module application
Arduino IDE	Used to develop and compile the application that run on Arduino Bluetooth Board
AVR Studio 4	Used to program the Atmeg328P microcontroller that is on Arduino Bluetooth Board

VII. Tracking Algorithm

Figure 4 depicts the layout of the environment used to develop the indoor locating system. As depicted, a single fixed, beacon node is installed in each room at a pre-determined location.

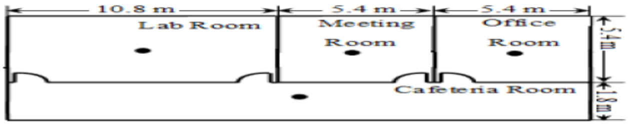

Fig.4. Layout of the overall system

The experiments were carried out using Arduino Bluetooth board [12] as beacons where there is a board installed per each room, and mobile phones from different brands (Nokia 5630 Music Express, BlackBerry 9700, and Samsung D900i) for measuring the Bluetooth signals. There is no software needed to pre-install on the mobile phones just it is required to be Bluetooth enabled and made discoverable. The physical environment consist of four rooms where there is typically three physical rooms titled as Office room, Meeting room, Lab room and finally the corridor along these rooms but virtually we considered it as another room named, Cafeteria room. The office room has size of 5.4x5.4 m^2 large, the meeting room has size of 5.4x5.4 m^2, the Lab room has size of 10x5.4 m^2 and the corridor has size of 21.6x1.8 m^2 large. The Locating Module run over the Arduino Bluetooth enabled boards/beacons and the Server Module run over a personal computer with a Bluetooth dongle [17]. It should be noticed that beacon nodes are placed such that once the mobile node is moved into a room it becomes closest to that room's beacon node, taking into account the effect of walls separating different rooms. This is rational to probabilistically which increase the RSSI value of the signal received from the closest beacon node over the values received from the others.

The server module will iterate periodically on all the boards, each iteration composed of two rounds, in the first round the server module will just send start "S" command to all the boards sequentially. In the second round, the server module might wait only once at the first board to finalize its inquiry time but all other following boards will be ready with its inquiry response and the server module will retrieve its data without waiting more time.

When the locating module receives the start, "S", command it will start the inquiry for nearby mobiles. Upon receiving this inquiry by nearby mobiles, each mobile node will reply that inquiry by an inquiry response that contains the MAC address of that mobile and its corresponding signal strength, RSSI.

The RSSI value in the inquiry response is automatically calculated by the Bluetooth MAC layer. The locating module will collect these data (MAC addresses and signal strength) and send it line by line to the server module just upon server module read operation. When the locating modules send all the data to the server module the locating module will be waiting for the next start command. The locating module will close the connection with the server module upon user request to exit/stop the application where the server module will send a stop command "X" to all beacon nodes that upon receiving the stop command will close the connection with the server module.

Now the server module processes the whole information and estimates the locations of all Bluetooth-enabled mobile phones. It then invokes the appropriate services for each mobile user according to the estimated location. Actually the Server module employ a trained neural network where for a certain mobile MAC address there will be a vector R that has the RSSI values of that MAC address for each room where RI refer to the RSSI value of that mobile as reported by the beacon node in room i. By applying this as an input to the trained neural network and based on the different values of the internal weights between neurons that reflect the environmental effect on the measured value. Some output neurons are activated referring to the estimated location (room-level). Figure 5 shows room adjacency relationships.

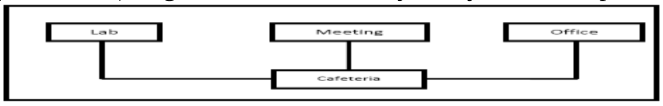

Fig.5. Room adjacency relationships

VIII. System Test Results

Figures 6, 7, 8 and 9 show total network error using eight, five, six and seven neurons respectively. Tables III, IV, V and VI show system positioning estimation accuracy using eight, five, six and seven neurons respectively. These results are taken through different attempts.

A. Attempt1

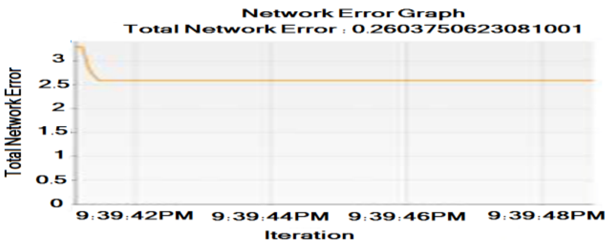

Fig.6. Total network error using eight neurons

Table III. System positioning estimation accuracy using eight neurons

Scheme	Refresh Rate	Accuracy
Neural network utilized	10 Sec.	73.93%
Room adjacency applied	10 Sec.	87.23%
Room adjacency applied and twice per room	20 Sec.	91.48%
Room adjacency and twice per room with confidence threshold applied	10 Sec.	91.48%

Using Neuroph, a neural network was created with 4 input neurons, 4 output neurons and the number of neurons in the hidden layer is equal to eight. Using the collected training data samples to train the neural network, and after testing we got in this attempt that the total network error is 26% as illustrated in figure 7.

B. Attempt 2

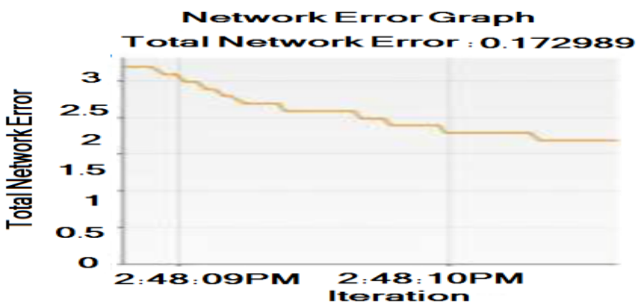

Fig. 7. Total network error using five neurons

Table IV. System positioning estimation accuracy using five neurons

Scheme	Refresh Rate	Accuracy
Neural network utilized	10 Sec.	82.97%
Room adjacency applied	10 Sec.	90.95%
Room adjacency applied and twice per room	20 Sec.	94.68%
Room adjacency and twice per room with confidence threshold applied	10 Sec.	94.68%

A new neural network was created with 4 input neurons, 4 output neurons and the number of neurons in the hidden layer is equal to five. Using the same collected training samples as previous attempt to train the neural network, after testing we got in this attempt that the total network error is 17% as illustrated in figure 8.

C. Attempt 3

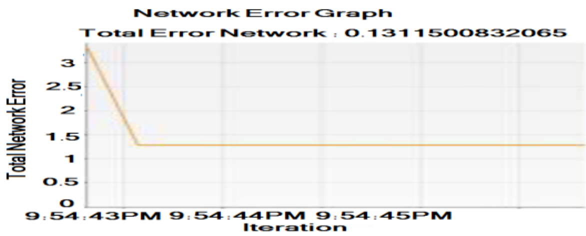

Fig. 8. Total network error using six neurons

Table V.System positioning estimation accuracy using six neurons

Scheme	Refresh Rate	Accuracy
Neural network utilized	10 Sec.	86.70%
Room adjacency applied	10 Sec.	95.21 %
Room adjacency applied and twice per room	20 Sec.	97.34%
Room adjacency and twice per room with confidence threshold applied	10 Sec.	97.34%

A new neural network was created with 4 input neurons, 4 output neurons and the number of neurons in the hidden layer is equal to six. Using the collected training samples, as in the previous attempts, to train the neural network, after testing we got in this attempt that the total network error is 13% as illustrated in figure 9.

D. Attempt 4

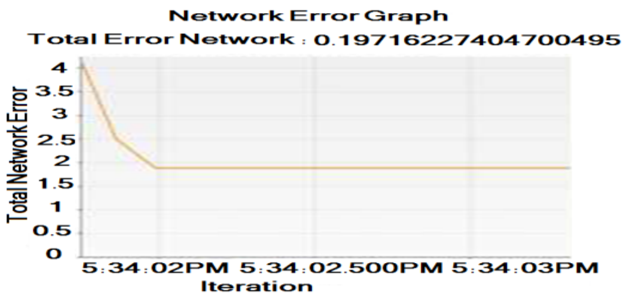

Fig.9. Total network error using seven neurons

Table VI. System positioning estimation accuracy using seven neurons

Scheme	Refresh Rate	Accuracy
Neural network utilized	10 Sec.	80.32%
Room adjacency applied	10 Sec.	89.63 %
Room adjacency applied and twice per room	20 Sec.	92.02%
Room adjacency and twice per room with confidence threshold applied	10 Sec.	92.02%

Finally, a new neural network was created with 4 input neurons, 4 output neurons and the number of neurons in the hidden layer is equal to seven. Using the collected training samples, as in the previous attempts, to train the neural network, after testing we got in this attempt that the total network error is 19% as illustrated in the figure 10. Finally table VII summarizes all the results concerning system positioning estimation accuracy for different number of neurons.

Table VII. System positioning estimation accuracy for different number of neurons

Scheme	**Accuracy as per the number of neurons in the hidden layer**			
	Five neurons	**Six neurons**	**Seven neurons**	**Eight neurons**
Neural network utilized	82.97%	86.70%	80.32%	73.93%
Room adjacency applied	90.95%	95.21 %	89.63 %	87.23%
Room adjacency applied and twice per room	94.68%	97.34%	92.02%	91.48%
Room adjacency and twice per room with confidence threshold applied	94.68%	97.34%	92.02%	91.48%

IX. CONCLUSIONS

As demonstrated, proper selection of the number of internal neurons in the hidden layer influences the accuracy of the results. From the mentioned trials we could conclude that the optimum size of the hidden layer in this typical environment is six neurons that gave the optimum results.

Another point utilizing room adjacency relationships consistently improved estimation accuracy. Also, to minimize the stochastic instability of the RSSI measurements, the user location is updated only when the system report two consecutive times the same decision. This effects on the system refresh rate and to retain the refresh rate again to its original value a confidence threshold value is applied. By this approach estimation accuracy and system stability are maximized. In a nutshell estimating location is based not only upon a single value reading of the RSSI from different rooms and selecting the highest RSSI. It based also on the nature of the underlying environment. The implicitly included in the neural network weights is calculated in the training phase and is applied on a post processing fixes to enhance system accuracy and stability.

References

[1] I.nat,Stat. “Bluetooth-enabled products poised to swim in the mainstream, ” April 2009.

[2] Y. X. Zhao, Q. Shen, and L. M. Zhang, “A Novel High Accuracy Indoor Positioning System Based on Wireless LANs,” Electromagnetics Research C, vol. 24, pp. 25-42, 2013.

[3] M. Ciurana, F. Barcel_o-Arroyo, and S. Cugno, “A robust to multi-path ranging technique over IEEE 802.11 networks,” *Wireless Networks,* vol.16,pp. 943-953, 2014.

[4] Miguel Dom´ınguez-Dur´an, D´ebora Claros, Cristina Urdiales, Francisco Coslado, and Francisco Sandoval, “Dynamic Calibration and Zero Configuration Positioning System for WSN, ” *The 14th IEEE Mediterranean Electrotechnical Conference, MELECON,* pp. 145-150, 5-7 May 2013.

[5] Hui Liu, Houshang Darabi, Pat Banerjee, Jing Liu, “Survey of Wireless Indoor Positioning Techniques and Systems, ” *IEEE Transactions On Systems And Cybernetics—Part C: Applications and Reviews,* vol. 37, no. 6, November 2014.

[6] John Krumm, Steve Harris, Brian Meyers, Barry Brumitt, Michael Hale, Steve Shafer, “Multi camera Multi-person tracking easy living , ” *Third IEEE International Workshop on Visual Surveillance, Dublin, Ireland,* pp.1-8, 2013.

[7] S. Gezici, Zhi T., G.B. Giannakis, H. Kobayashi, A.F. Molisch, H.V. Poor, and Z. Sahinoglu. “Localization via ultra-wideband radios: a look at positioning aspects for future sensor networks,” *Signal Processing Magazine, IEEE,* vol.22,issue 4, pp.70-84, 2013.

[8] G. Zhang, S. Krishnan, F. Chin, and C.C. Ko., “UWB multicell indoor localization experiment system with adaptive TDOA combination, ” *In Vehicular Technology Conference, IEEE 68th,* pp. 1-5, 2013.

[9] Keijer, U., “With IT at Home – an Evaluation of a Three-year Project,” *TRITA-ARK-Research Publication 2007:4, Royal Institute of Technology, Architecture and the Built Environment, Stockholm, Sweden,* 2014.

[10] Ambili Thottam Parameswaran, Mohammad Iftekhar Husain, and Shambhu Upadhyaya, “Is RSSI a Reliable Parameter in Sensor Localization Algorithms – An Experimental Study, ” *Department of Computer Science and Engineering, State University of New York at Buffalo, USA,* pp. 1-5, 2014.

[11] Ward, A., Jones, A., and Hopper, A., “A new location technique for the active office, ” *IEEE Personal Communications,* vol.4, pp. 213-222, 2012.

[12] Barlow, J. and Gann, D., “A changing sense of place: Are integrated IT systems reshaping the home?, ” SEWPS No 18, SPRU, University of Sussex, England, 2005.

[13] Lansley, P., “The promise and challenge of providing assistive technology to older people, ” Age & Ageing, pp. 439-440, 2011.

[14] Robert J. Orr, Gregory D. Abowd, “The Smart Floor: A Mechanism for Natural User Identification and Tracking, GVU Technical Report GIT-GVU-00-02 (full paper), ” January 2009.

[15] M. Montemerlo, J. Pineau, N. Roy, S. Thrun, and V. Verma, “Experiences with a mobile robotic guide for the elderly,” *In Eighteenth national conference on Artificial intelligence, Canada,* 2012.

[16] Joshua Correa, Ed Katz, Patricia Collins, Martin Griss, “Room-Level Wi-Fi Location Tracking, ” *Carnegie Mellon Silicon Valley, technical report: MRC-TR-2008-02,* November 2012.

[17] K. Pahlavan, X. Li, and J. Makela, “Indoor geolocation science and technology,” *IEEE Commun. Mag.,* vol. 40, no. 2, pp. 112–118, Feb. 2013.

[18] Janire Larranaga, Leire Muguira, Juan-Manuel Lopez-Garde, and Juan-Ignacio Vazquez, “An Environment Adaptive ZigBee-Based Indoor Positioning Algorithm, ” *IEEE International Conference on Indoor Positioning and Indoor Navigation, Switzerland,* pp.1-8, 15-17 September, 2013.

[19] Gann, D., Barlow, J. and Venables, T., “Digital Futures: Making Homes Smarter,” *Chartered Institute of Housing, England,* 2009.

[20] Bahl, Paramvir and Venkata N. Padmanabhan, “RADAR: An In-Building RF-based User Location and Tracking System.” *Proceedings of IEEE Infocom 2007, Tel-Aviv, Israel,* pp.217-230, March 2007.

[21] A. Harter and A. Hopper, “A Distributed Location System for the Active Office, ” *IEEE Network,* January 2007.