Wireless Based Object Tracking Based on Neural Networks

Kurt Derr
Idaho National Laboratory
2525 Freemont Avenue
Idaho Falls, ID 83415, USA
derr5843@uidaho.edu
Kurt.Derr@inl.gov

Milos Manic
Department of Computer Science
University of Idaho at Idaho Falls
1776 Science Center Dr., Ste.306
Idaho Falls, ID 83402, USA
misko@uidaho.edu

***Abstract*— Location Based Services (LBS), context aware applications, and people and object tracking depend on the ability to locate mobile devices, also known as *localization,* in the wireless landscape. Localization enables a diverse set of applications that include, but are not limited to, vehicle guidance in an industrial environment, security monitoring, self-guided tours, personalized communications services, resource tracking, mobile commerce services, guiding emergency workers during fire emergencies, habitat monitoring, environmental surveillance, and receiving alerts.**

This paper presents a new neural network approach (LENSR) based on a competitive topological Counter Propagation Network (CPN) with k-nearest neighborhood vector mapping, for indoor location estimation based on received signal strength. The advantage of this approach is both speed and accuracy. The tested accuracy of the algorithm was 90.6% within 1 meter and 96.4% within 1.5 meters. Several approaches for location estimation using WLAN technology were reviewed for comparison of results.

***Keywords*—RSS, localization, neural network, CPN, k-nearest neighbor, signature recognition, GPS.**

I. INTRODUCTION

Both indoor and outdoor location estimation is a significant problem posing serious technical challenges. Location estimation, also known as localization, concerns the positioning of mobile devices in some physical space. Although location estimation represents an active area of research, proposed solutions are generally cost prohibitive, inaccurate, or infeasible due to practical issues.

GPS and wireless technologies are useful for determining location. Although GPS supports LBS, the number of mobile devices using GPS technology available in the marketplace today is limited. This is due to high cost, power requirements, and the inability to work in certain environments, such as indoors, underground, and in city canyons (an area of a city with narrow streets and high buildings) [1, 2].

This paper focuses on indoor location estimation from radio signal strength (RSS) values received by a mobile device with wireless (WiFi) capabilities as the device moves around an area of a building; e.g., a smart phone, personal digital assistant, equipment or packages with WiFi sensors, or a robot with WiFi capabilities. The ubiquity and low cost of 802.11 technology makes localization based on wireless local area network (WLAN) technology a viable alternative to GPS, enhancing the value of the wireless network.

The building has wireless access points (APs) acting as *anchors* deployed at various locations. First, a radio grid map is obtained offline. This way attenuation and reflection of signals in an urban environment is recorded as it is, resulting in both computational time savings and precision of recorded signals. The mobile device then estimates its position algorithmically using RSS values received from access points and the grid map. This is entirely a client-based system in that the mobile device does not send packets to a server in order to determine the location of the device. WiFi coverage of the complete area of interest is necessary for accurate location estimation.

In this paper we present a new Counter Propagation neural Network (CPN) with k-Nearest Neighbor (k-NN) algorithm for location estimation using WiFi received signal strength. To the best of our knowledge this algorithmic approach has not been previously used for location estimation. The paper is organized as follows. Section 2 discusses related work, section 3 reviews the principles of mobile device location estimation, section 4 describes the proposed CPN with k-NN algorithm, section 5 presents test results computed in a MATLAB environment, and section 6 states our conclusions.

II. RELATED WORK

Wireless localization schemes are generally categorized as deterministic or probabilistic [3]. The deterministic techniques are range or proximity based. The range based approach uses the characteristics of the channel, such as Received Signal Strength (RSS), to find the distance from a mobile device to wireless access points. Alternatively RSS fingerprinting techniques may used to locate a mobile device in a building. Neural networks, specifically a generalized regression neural network, have been used as the pattern matching algorithm in geo-location systems [4].

The probabilistic technique [5,6,7,8,9] constructs a conditional probability distribution over some area of interest to determine the likelihood of a mobile device being at some position at a specific point in time. Probabilistic techniques are computationally more expensive than deterministic techniques but provide a higher degree of accuracy (90% within 2 meters [10]).

Some examples of a wireless localization systems and techniques follow. These systems and techniques typically have an offline training phase and an online location determination phase [11]. Some location estimation techniques build a radio map in the offline phase that represents the RSS values to each reachable AP from every location in the area of interest. A location estimation algorithm run on a mobile device requires a copy of the radio map. The advantage of running the algorithm on the mobile device is the preservation of user privacy and improved scalability.

RADAR [12, 13] uses RSS measurements obtained from multiple locations to triangulate a user's position in an area. An experimental radio map is built offline by taking measurements in all possible grid locations of the area of interest. The system performs both location estimation and user tracking to within 2 to 3 meters of the actual location.

The Clustering and Decision Tree-based [14] method (CADET) selects the set S of wireless access points that give the best performance for each location of a wireless area in the offline phase. The grid space is then partitioned into clusters and a decision tree is built for each cluster. In the online mode the RSS values from selected access points are used to determine which cluster and coarse location is associated with the device. Next, the decision tree for the identified cluster is evaluated resulting in a specific grid location. The best accuracy of CADET is 83.4% within 1.5 meters.

The Joint Clustering (JC) [10] technique uses clustering of location maps and probability distributions. A cluster, which represents a set of locations that shares a common set of access points, is calculated offline, as well as the joint probability distributions of the signal strength of different access points. During the online location estimation phase RSS values are acquired from some set of APs , which are used to determine the cluster to search for the probable location. The radio map and Baye's theorem are used to determine the most probable user location within a cluster. The accuracy of this system is 90% to within 7 feet.

Uncertainty in RSS signal measurements can be modeled as fuzzy sets. [15] divides the area of interest into zones. A radio map is developed offline and is used to train the fuzzy inference system. There are six fuzzy sets for RSS: Excellent, Very Good, Good, Low, Very Low, and None. The degree of membership of a mobile device to a specific area is used to determine the location estimate, providing an accuracy of near 90%.

A generalized regression neural network (GRNN) is used for a pattern matching algorithm in [4]. A measured RSS value for each AP and the corresponding grid location in the radio map are inputs to the neural network during the training phase. The GRNN has one hidden layer, and an output layer corresponding to two neurons representing the *x* and *y* locations in the radio map. During the online phase a set of RSS values are input to the GRNN and the output is the estimated user's location in *x* and *y* grid map coordinates. The maximum error between estimated and true positions for the test data was 43.2 meters. The location accuracy for the test data is 45% to within 5 meters.

III. MOBILE DEVICE LOCATION ESTIMATION

The approach to mobile device location estimation presented in this paper is based on a comparison of RSS signal vectors recorded by a mobile device and RSS vectors from a radio grid map. The radio grid map can be created offline in two ways. One way is either to have a person manually collect and record RSS signal strength values for each grid location, or to do this task automatically with a robot. With this approach, both precision (actual signal is recorded) and computational time savings are achieved (no analytic determination of attenuation and reflection is needed). Another way is the creation of a theoretical propagation model representing the RSS signal levels that are calculated for every location of the radio map based on propagation equations. For the sake of simplicity, the latter approach is taken in this paper. The effectiveness of the presented algorithm is the same regardless of the way in which the radio grid is created. The advantage of a recorded map is that complex, analytic modeling [16] of signal attenuation and reflection in an indoor environment can be effectively avoided, resulting in more correct, actual radio map. The use of a theoretical model allows the algorithm to be deployed in a new environment without having to physically acquire signal strength readings for building a radio map. Alternatively, the radio map may be constructed by 1) manually recording RSS readings from access points in each designated grid location in the area of interest, or 2) using an autonomous vehicle/robot to collect signal strength information. The CPN with k-NN AP approach to location estimation is valid regardless of how the radio map is built. Locations and WiFi transmitters and receivers power are all that is needed to build the radio map. The APs are expected to be homogeneous; i.e., same transmitter power. Cable and connector losses are omitted to simplify the model. The theoretical model is based on the following equations for received signal strength R_x and path loss L_p [17]:

$$R_x = T_x - L_p \quad (1)$$

where R_x is received signal strength value. T_x is AP power in dB, and L_p is path loss, and

$$L_p = 33dB + N * \log_{10}(D) + 20 * \log_{10}(f) \quad (2)$$

where f is a frequency in gigahertz. N is a path loss exponent, and D is a distance in meters.

Based on equations (1) and (2), for factory environment path loss and access point power (N = 5.5, T_x = +20dBm), and AP frequency f = 2.4, received signal strength R_x is:

$$R_x = 20 - 33 - 5.5 \log_{10}(D) - 20 * \log_{10}(2.4) \quad (3)$$

Equation (3) shows how the signal strength decreases exponentially with distance from a WiFi access point regardless of transmission power and antenna gain.

The theoretic model is developed by using equation (3) for calculating the received signal strength for each access point in every location of the radio map. Some test values are calculated by adding noise to the theoretical model. The theoretical map of RSS values for a 20 meter square area (*n* ***x*** *m* cells) with 4 WiFi access points is shown in Figure 1 for each individual access point.

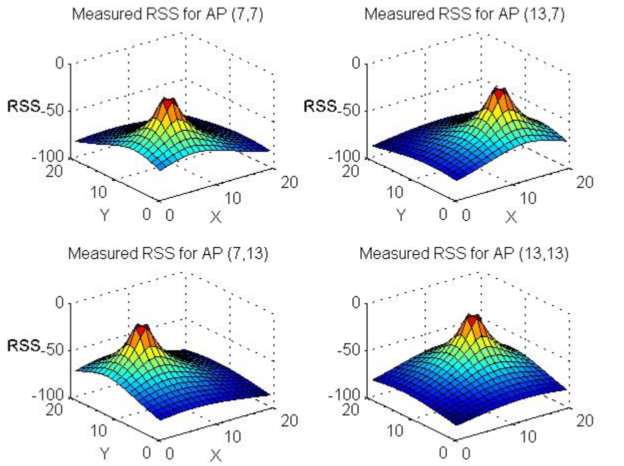

Figure 1. RSS Landscape per Access Point

IV. Counter Propagation Network (CPN) with K-Nearest Neighbor

A feed-forward CPN network generally consists of two layers. This modular neural network [18, 19] uses other neural networks as building block components and provides capabilities that a single monolithic neural network is not capable of providing [20].

The presented approach (LENSR) is based on a symbiotic algorithm of a CPN and k-nearest neighbor approach with multiple winning neurons in the first layer (see Figure 2). CPN networks were introduced by Hecht-Nielsen in late 1980s as two layer, vector-to-vector hetero-associative memory networks [21, 22, 23]. This network was selected for a number of advantages. CPNs are very fast and easy to use: training is performed by design, hence training is much faster than a typical 2-layer network; there is no feedback or delay during recall mode.

In addition to being simple to use and control, these networks can recognize patterns that have not been seen by the network before. In the case of no perfect matching, the network will find the closest match possible and provide the output accordingly. In the case of multiple closest matches, the network will provide a composite output, proportional to the degree of similarity between unseen and previously stored patterns. Input vector recognition is done simultaneously against all previously learned vector patterns. The advantageous speed of these networks is a consequence of the parallelism, inherent to topologic organized CPNs.

The area of interest is divided into a grid of n ***x*** m cells as shown in Figure 1. Each access point, plotted as a hump in the diagram, produces an RF signal detectable by other WiFi devices within radio range. Each RSS signature is a k-dimensional vector with signals measured from each of k access points. These *Signal Space Vectors* (SSV) are augmented with the recorded spatial coordinates for each specific cell (x,y):

$$SSV_i \overset{def.}{=} (AP_1, AP_2, ..., AP_k, x_k, y_k),\ i = 1,2,...,n \cdot m \tag{9}$$

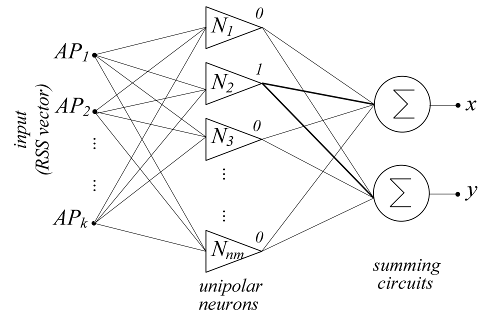

Figure 2. Counter Propagation Network

Such SSV are used as training points in the presented algorithm. In an indoor/outdoor environment these signals may be measured and collected over an area of interest manually or by an autonomous vehicle.

The Kohonen layer in Figure 2 has one neuron for each location in the n ***x*** m grid. There are four inputs to each neuron as represented by the RSS values from each access point. For each access point in the area of interest the number of inputs to each neuron increases by 1. The outputs of the Kohonen layer are weighted coordinate locations. By design, the connections between the Kohonen neurons and Grossberg summing circuits are the weights, w_i, based on the proportional normalized distance vectors. The Grossberg layer, which is two summing circuits, performs the summing of the outputs from the Kohonen layer and produces both an x and y coordinate value as outputs.

The presented LENSR algorithm goes as follows.

Step 1: Collect measured SSVs for each location in the grid map. Retain SSVs in a table T_1.

Step 2: As a mobile device moves through the area of interest the device records RSS values at its current location. This represents a test RSS vector in signal space. The test vector is used to determine location based on the training data (radio map).

Step 3: Compare test vector to each measured vector in table T_1. This is done in one step due to the inherent parallelism of the used CPN network. **Step 3.1:** If test vector equals any measured vector, then return location (x,y).

Step 3.2: Else calculate distance between RSS measured signal vectors and test vector and retain in table T_2. Retain all measured vector locations in table T_3 for distances satisfying the minimum distance threshold.

Step 4: Calculate the inverse normalized proportion of distances in table T_2 as noted above, producing weights w_i.

Step 5: Use weights, w_i, and the measured x and y locations from Table T_3 to calculate the output of the CPN, $O_{(x,y)}$ as in equation 8. $O_{(x,y)}$ is the cumulative location representing the test vector location, (x,y). Error is calculated as follows:

$$Error = \sqrt{(p_x - a_x)^2 + (p_y - a_y)^2}$$

where p is a predicted location and a is the actual location.

The LENSR algorithm presented in this paper can recognize an unseen RSS signature by matching it against the table of RSS vectors measured previously. If an exact match is found, the corresponding location or (x,y) pair is returned. If the exact match is not found, a number of similar patterns (patterns within certain distance threshold) are returned. The minimum distance threshold is defined as a cell size (the test examples section presents results with various threshold size and various distance metrics). Further, the inverse normalized proportion of distances is used to produce a cumulative location of unseen pattern in following manner:

1. Calculate distance between radio map RSS signal vectors and test vector. This is computing a distance in the signal space to other samples based on a distance metric.

If distance is within a previously defined threshold, record distance and vector coordinates. The result of this comparison against all RSS vectors from a radio map is: $d_i = (d_{i1}, d_{i2}, ..., d_{ik})$

2. Calculate weights, w_i, (proportional normalized distances) of these nearest neighborhood vectors as:

$$w_i = \frac{\left(\frac{1}{d_i}\right)}{\sum_{i=1}^{k}\left(\frac{1}{d_i}\right)} \text{ , } i=1,1,...,k \tag{4}$$

$$\sum_{i=1}^{n} w_i = 1 \tag{5}$$

where d_i is a distance to i^{th} neighborhood vector, and k is the number of nearest neighbors. Distance metrics may be the Euclidean, Manhattan, or other. The Euclidean distance between (x_1,y_1) and (x_2,y_2) is defined as:

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \tag{6}$$

while the Manhattan distance is defined as:

$$d=| x_1-x_2| + | y_1-y_2|. \tag{7}$$

The sum of all weights, w_i, equals 1. This approach to calculating of weights ensures that the neighbors that are closer in signal space get weighted more heavily than those far away.

3. Calculate test vector location as the output of the CPN, $O_{(x,y)}$.

$$O(x,y) = \sum_{i=1}^{n} (w_i * p_i) \tag{8}$$

where $p_i = (x_i,y_i)$ is the location of neighborhood vector.

V. TEST RESULTS

Several tests were performed to access the accuracy of the model. Test 1 used a small number of test points so the effect of RF signal attenuation on nearest neighbors could be visualized. Tests 2 and 3 used test points at x mid-grid locations (1.5, 2.5,...., 20.5) along each y point (1 to 20). Test 4 used 8 access points to calculate the theoretic map. Test 5 used the Manhattan distance metric. Tables 1 and 2 summarize the test results.

Test 1:

Input: 5 unique test SSVs different from measured SSVs,
Output: 4 calculated positions with their neighborhood plots
Discussion: The calculated neighborhoods and predicted test positions are shown in Figure 3 with access points marked with an X. An estimated location is not available for one of the test vectors because no neighbors were within the threshold. The symmetry of the neighbors around one side of the calculated location is due to the similarity of RSS vectors within the distance threshold. Neighbors do not appear on the other side of the calculated location because the measured RSS values increase as the access point is approached. This effectively increases the distance to the access point, falling outside the threshold boundary and resulting in no neighbors for this area.

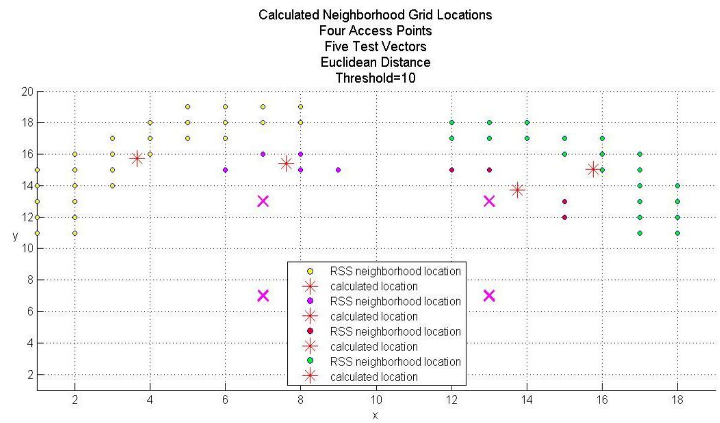

Figure 3. Neighborhoods and Estimated Positions for Five Test Points

Test 2:

Input: 400 unique test SSVs different from measured SSVs,
Output: Calculated positions for each test SSV
Discussion: Several thresholds for k-NN were tested with the smallest threshold yielding the best results in terms of cumulative error. Error increases, as expected, as the distance between RSS test and measured SSVs increases.

Locations at the periphery of the grid map have significantly higher error (see Figure 4) due to distance from the closest access point. The tested accuracy of the algorithm was 90.6% within 1 meter and 96.4% within 1.5 meters at a threshold of 6.

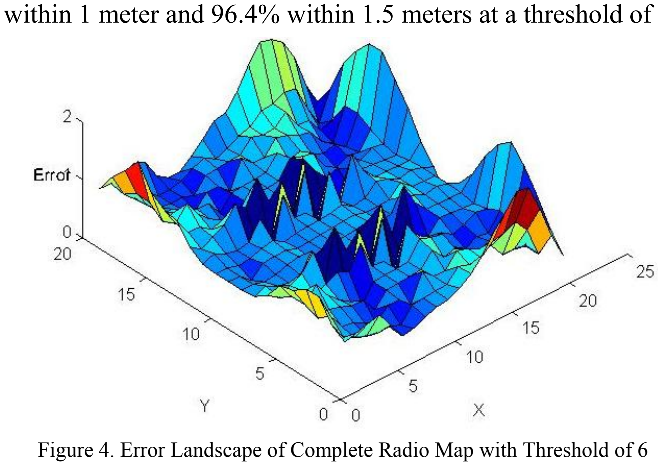

Figure 4. Error Landscape of Complete Radio Map with Threshold of 6

Some spatially close neighbors to the RSS test vectors provide the optimal data for estimating the position of the mobile device. As the distance to the neighbors increases the error also increases due to the inclusion of points further away, which are inaccurate. Some signal strength components of the RSS vector will increase while others will decrease as a mobile device moves across the landscape. Therefore even some test points close to the access points, which measured RSS values are based on, will have a higher error rate than others. Figures 5 and 6 plot the predicted test locations where the error was less than or equal to 1, and the thresholds were 6 and 15, respectively. As the threshold is increased, more RSS vectors will fall within the threshold and be used to calculate the predicted location of the mobile device. This effectively increases the error in the calculation since some of those vectors are close in signal space but not in spatial distance.

Test 3:

Input: Same test locations as test 2 minus the test points on the periphery of the grid map.

Output: Calculated positions for each test SSV

Discussion: The error landscape using a distance threshold of 6 and 15 are shown in Figure 7 and 8, respectively. Signal strength from an access point decreases exponentially with distance. Therefore points closer to an access point will provide a more accurate relationship between distance and signal strength. The number of measured RSS vectors used to calculate the location of the mobile device increases with threshold, which increases the error. The error rate between access points (dark blue peaks) in Figure 8 is relatively constant indicating that measured RSS vectors in this region have similar values. The tested accuracy of the algorithm was 94.5% within 1 meter and 100% within 1.5 meters at a threshold of 6.

Test 4:

The density of the access points for the radio map was increased from 4 per 20 square meters to 8 and tests 2 and 3 were rerun. The best tested accuracy of the algorithm was 96.9% within 1 meter and 100% within 1.5 meters at a threshold of 6.

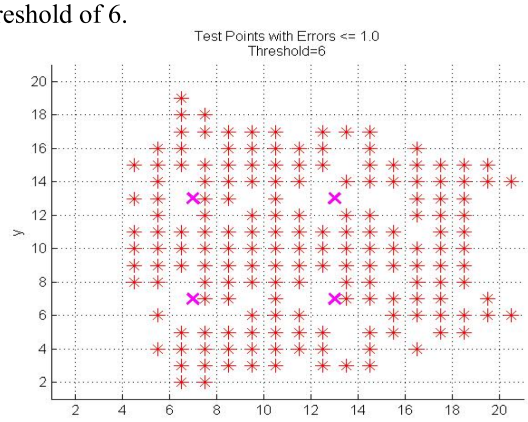

Figure 5. Test Points with Threshold = 6, (X = Access Point)

Test 5: Test 2, 3, and 4 were rerun using the Manhattan distance metric.

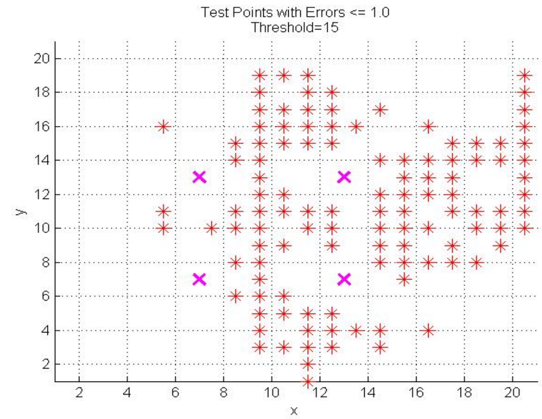

Figure 6. Test Points with Threshold = 15

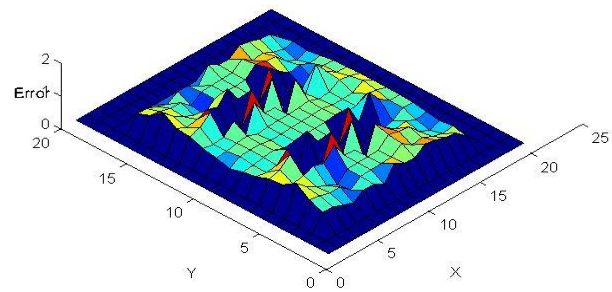

Figure 7. Error Landscape with Threshold of 6

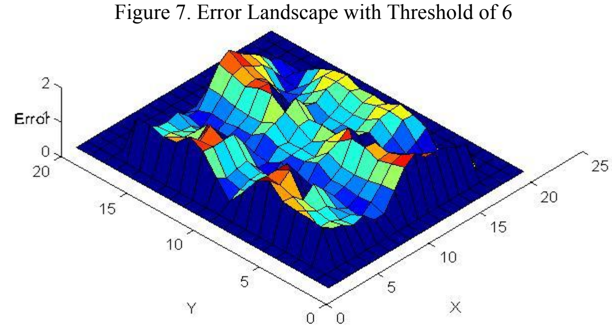

Figure 8. Error Landscape with Threshold of 15

Table 1 summarizes the accuracy of the LENSR approach for location estimation using the Euclidean distance metric, a nearest neighbor value, or threshold *T*, of 6 and 15, and both 4 and 8 access points. The accuracy of the algorithm is calculated for predicted test point locations with less than or equal 0.5, 1.0 and 1.5 meters error. A threshold, *T*, of 6 produces the best accuracy for 0.5 and 1.0 meters. The accuracy of the CPN algorithm, using the Euclidean distance metric, decreases with increasing threshold. The accuracy of the CPN algorithm increases with additional access points.

The results of the same test using the Manhattan distance metric is shown in Table 2. The Manhattan distance metric will produce a larger distance value for grid locations that are located spatially diagonally from one another. The greater the distance is between neighborhood grids the greater the error.

The CPN approach compares favorably to RADAR, CADET, Fuzzy Location, Joint Clustering Technique, and RSS Fingerprinting with Neural Network.

# Test Points for *T<=6*	# Access Points	Error <= 0.5 meter	Error <= 1.0 meter	Error <= 1.5 meter
392	4	41%	90.6%	96.4%
217	4	46%	94.5%	100%
366	8	43.7%	95.6%	100%
191	8	45.6%	96.9%	100%
# Test Points for *T<=15*				
392	4	19.1%	54.6%	85.5%
217	4	24.9%	65.9%	93.6%
384	8	27.9%	78.9%	94%
191	8	34.5%	88%	99.5%

Table 1. LENSR Accuracy: Threshold = 6/15, Euclidean Distance Metric

# Test Points for *T<=6*	# Access Points	Error <= 0.5 meter	Error <= 1.0 meter	Error <= 1.5 meter
387	4	5.2%	34.4%	51.4%
212	4	9%	36.3%	52.4%
201	8	27.4%	87%	97.5%
79	8	25.3%	78.5%	96.2%
# Test Points for *T<=15*				
395	4	4.6%	16.7%	32.2%
220	4	10%	25.9%	44.1%
386	8	20.7%	82.6%	96.1%
211	8	19%	84.4%	97.6%

Table 2. LENSR Accuracy: Threshold = 6/15, Manhattan Distance Metric

VI. Conclusions

The CPN is a viable architecture for calculating location based on RSS values. The advantages of the CPN for location estimation are multifold. The first is the inherent parallelism of RSS signature matching due to the intrinsic parallelism of the CPN architecture. Another advantage is that the actual signal distribution is recorded as it is. Complex signal deflection and attenuation calculations are avoided and hence the precision of algorithm is increased. The third advantage is the computational inexpensiveness of LENSR, an elegant CPN based pattern matching and linearization of obtained neighboring locations.

The LENSR algorithm has a higher accuracy for location estimation than other neural network, nearest neighbor, and clustering approaches reviewed in this paper. Accuracy increases, as expected, as the density of access points is increased for the radio map area of coverage.

VII. References

[1] R. Galvin, “Anchor Position Estimation For Use In Ad Hoc Wireless Sensor Localisation,” Masters Thesis, Open University, March 11, 2007.

[2] G. Borriello, M. Chalmers, A. LaMarca, P. Nixon, “Delivering Real World Ubiquitous Location Systems”, Communications Of The ACM March 2005/Vol. 48, No. 3, pp. 36-41

[3] J. Yin, Q. Yang, L. Ni , “Adaptive Temporal Radio Maps for Indoor Location Estimation,” Proceedings of the 3rd IEEE Int’l Conf. on Pervasive Computing and Communications (PerCom 2005)

[4] C. Nerguizian, C. Despins, S. Affes , “Indoor Geolocation with Received Signal Strength Fingerprinting Technique and Neural Networks,” ICT 2004, LNCS 3124, pp. 866–875, 2004.

[5] R. Peng, M. Sichitiu, “Probabilistic Localization for Outdoor Wireless Sensor Networks,” Mobile Computing and Communications Review, Volume 11, Number 1, pp. 53-64, January 2007

[6] S. Ren, Q. Li, H. Wang, X. Chen, X. Zhang , “A Study on Object Tracking Quality under Probabilistic Coverage in Sensor Networks,” Mobile Computing and Communications Review, Volume 9, Number 1, pp. 73-76, January 2005.

[7] D. Fox, J. Hightower, L. Liao, and D. Schulz, “Bayesian Filtering for Location Estimation,” IEEE Pervasive Computing, vol. 2, no. 3, pp. 24-33, 2002.

[8] V. Olivera, J. Plaza, O. Serrano, “WiFi Localization Methods for Autonomous Robots,” Robotica, Volume 24 , Issue 4 (July 2006), Pages: 455 – 461

[9] F. Wang, L. Qiu, S. Lam, “Probabilistic Localization for Outdoor Wireless Sensor Networks,” Mobile Computing and Communications Review, Volume 11, Number 1, pp. 3-14, January 2007

[10] M. Youssef, A. Agrawala, and U. Shankar, “WLAN Location Determination via Clustering and Probability Distributions,” Proc. IEEE Pervasive Computing, pp. 143-152, Mar. 2003.

[11] A. Chen, C. Harko, D. Lambert, P. Whiting, “An Algorithm for Fast, Model-Free Tracking Indoors,” Mobile Computing and Communications Review, Volume 11, Number 3, pp. 48-58, July 2007

[12] P. Bahl, V. Padmanabhan, “RADAR: An In-Building RF-Based User Location and Tracking System,” *IEEE INFOCOM2000 Conf.,* pp. 775-784

[13] K.Whitehouse, C.Karlof, D.Culler, “A Practical Evaluation of Radio Signal Strength for Ranging-based Localization”, Mobile Computing and Communications Review, Volume 11, Number 1, pp. 41-52, January 2007.

[14] Y. Chen, Q. Yang, J. Yin, X. Chai , “Power-Efficient Access-Point Selection for Indoor Location Estimation,” IEEE Transactions On Knowledge And Data Engineering, Vol. 18, No. 7, July 2006

[15] J. Astrain, J. Villadangos, J. Garitagoitia, J. Mendivil, V. Cholvi, ”Fuzzy Location and Tracking on Wireless Networks,” MobiWAC ’06, October 2006, pp. 84-91

[16] H. Hashemi, “The Indoor Radio Propagation Channel,” vol. 81,no. 7, pp. 943-968, 1993.

[17] Wireless Mesh Network for Building Automation, Industrial Wireless Book Issue 10:2, http://wireless.industrial-networking.com/articles/articledisplay.asp?id=1264

[18] K. Derr, M. Manic, “Intelligent Control in Automation Based on Wireless Traffic Analysis”, 12th IEEE International Conference on Emerging Technologies and Factory Automation, Patras, Greece, September 2007.

[19] R. Jang, Neuro-Fuzzy Modeling: Architectures, Analyses and Applications, PhD Thesis, University of California, Berkeley, July 1992.

[20] P.Melin, O. Castillo, “Hybrid Intelligent Systems for Pattern Recognition”, IJCNN, Tutorial T4_3, Orlando, Florida, August 12-17, 2007

[21] R. Hecht-Nielsen, "Nearest Matched Filter Classification of Spatio-temporal Patterns", special report published by Hecht-Nielsen Neuro-Computer Corporation, San Diego, California, June 1986.

[22] R. Hecht-Nielsen, "Counter-Propagation Networks", IEEE First International Conference on Neural Networks, Volume II, 1987.

[23] R. Hecht-Nielsen, "Neurocomputing: Picking the Human Brain", IEEE Spectrum 25(3), March 1988, pp. 36-41.