SEARCH SPACE REDUCTION IN DCM POSITIONING USING UNSUPERVISED CLUSTERING

Rafael Saraiva Campos[1], *Lisandro Lovisolo*[2] *and Marcello L. R. de Campos*[1]

1-PEE/COPPE/UFRJ, 2-DETEL/FEN/UERJ
rafaelsaraivacampos@gmail.com, lisandro@uerj.br, mcampos@ieee.org

ABSTRACT

Database correlation methods (DCM) are used to locate mobile stations (MS's) in wireless networks. A target radio-frequency (RF) fingerprint - measured by the target mobile station - is compared with georeferenced RF fingerprints, previously stored in a correlation database (CDB). In this paper, two unsupervised clustering techniques (K-medians and Kohonen Layer) were applied to reduce the search space inside the CDB. The clustering effects on the computational cost of the positioning method and on the positioning accuracy were experimentally evaluated using 46200 target fingerprints and a CDB with 924 reference fingerprints, containing Received Signal Strength (RSS) values of 136 WiFi 802.11b/g networks in a 12-floor building. A reduction of 81% in the average time to produce a position fix was observed, as well as a 38% decrease in the DCM average positioning error and a 6% improvement in the floor identification accuracy.

I. INTRODUCTION

The issue of MS's positioning in wireless networks has been receiving growing attention, from both commercial and regulatory point of views. This is even more noticeable in cellular telephony networks, where the the Federal Communications Commission (FCC) has specified a minimum accuracy requirement for the positioning of MS's originating emergency calls [1]. There is a growing number of MS's equipped with built-in Global Positioning System (GPS) receivers. In open areas, GPS yields the highest location precision, but is usually unavailable in indoor environments [2]. In this scenario, RSS based location techniques are used both in cellular and WiFi networks. However, in such environments, positioning using WiFi RSS values yields higher precision, due to the usually higher density of WiFi access points (AP's) in a indoor environment, in relation to cellular micro or even picocells. This, coupled with the increasing number of cellular MS's equipped with built-in WiFi receivers, makes WiFi RSS based positioning a viable choice for cellular MS location in indoor environments.

RSS based DCM is a viable alternative for indoor WiFi positioning [3]. There is a wide variety of DCM solutions in the literature, but all share the same basic elements [4] [5]. One of these elements is the CDB search space reduction technique, which has an impact both on the method's computational complexity - and therefore on the time required to produce a position fix - and on the positioning precision. In this paper, two techniques of unsupervised clustering - K-medians and Kohonen layer - are used to reduce the search space within the CDB, and the effects on the time to produce a position fix and on the location precision are experimentally evaluated.

The remainder of this work is organized as follows: Section II introduces the basic elements of DCM; Section III provides an overview of the proposed solution; Section IV describes the applied clustering techniques and justifies the use of unsupervised training; Section V details the experimental evaluation; and Section VI brings a brief conclusion.

II. DATABASE CORRELATION METHODS

DCM, also known as RF fingerprinting positioning, is a class of MS positioning methods that can be applied in any wireless network. Even though there is a wide variety of such methods, all present the same basic elements: RF fingerprints, a CDB, techniques to reduce the search space within the CDB, and the correlation of RF fingerprints.

II-A. RF Fingerprint

A RF fingerprint is a set of RF signal parameters. Those parameters are measured by the MS or by its anchor cells. Just like a human fingerprint, which carries the unique identification of a person, a RF fingerprint is expected to uniquely identify a geographic position.

A RF fingerprint can be classified as either a *target* (**TFing**) or *reference* (**RFing**) fingerprint. A **TFing** is the RF fingerprint associated with the MS which is to be localized, i.e., it contains signal parameters measured by the MS or by its anchor cells. The **RFing**'s are the RF fingerprints collected or generated during the training phase and stored in the CDB. Each **RFing** is associated with a 3-uple of geographic coordinates (x, y, z). The fingerprint structure used in this work, common to both **RFing**'s and **TFing**'s, is defined by the vector $\vec{F} = [\text{RSS}_1 \ldots \text{RSS}_N]$, where N is a constant which informs the number of WiFi 802.11b/g networks used in the position fix.

II-B. Correlation Database

The CDB is the set of **RFing**'s. The CDB is built during the DCM training phase [6], using radio propagation modeling, field measurements or a combination of both [7]. Each CDB entry is described by $\left(\vec{F}, x, y, z\right)$, where $\vec{F}$ is the **RFing** associated to the point defined by coordinates (x, y, z).

II-C. Techniques to Reduce the Search Space within the CDB

The search space or correlation space is a subset of the **RFing**'s stored in the CDB. The **RFing**'s in this subset are compared to the **TFing** to locate the MS. The geographic coordinates associated with the **RFing**'s in the search space are candidate solutions for the MS positioning problem. The CDB might be quite large and analyzing all RF fingerprints stored in it might be very time consuming. Therefore, all fingerprinting location techniques apply some method to reduce the search space within the CDB. As a consequence, the time required to produce a position fix is also reduced. Some of the techniques applied in the literature are deterministic filtering [8] and optimized search using genetic algorithms [9], both applied upon RSS maps built with empirical propagation models [10]. In [11], the search space is reduced by clustering the measurement points, i.e, the candidate solutions. This clustering is based on the identity of the q WiFi networks with the highest RSS at each measurement point.

II-D. Correlation of RF Fingerprints

The MS is assumed to be located at the point whose **RFing** has the highest correlation or similarity with the **TFing**. Comparison of the **TFing** and **RFing**'s can be carried out by calculating the distance between these fingerprints in the N-dimensional RSS space [8] or using back-propagation artificial neural networks (ANN's) with supervised training [12].

If $\boldsymbol{C}$ is the set of **RFing**'s in the CDB and $\boldsymbol{D}$ is the set of **RFing**'s in the search space, then $\boldsymbol{D} \subseteq \boldsymbol{C}$. Let $\vec{F}$ be the **TFing** measured by the MS to be localized. The **RFing** $\vec{F}_k$ most similar to $\vec{F}$ is given by:

$$k = \underset{i}{argmax} \left[-\left(\vec{F} - \vec{F}_i\right)\left(\vec{F} - \vec{F}_i\right)^{\mathrm{T}} \right], \; \forall \vec{F}_i \in \boldsymbol{D} \tag{1}$$

The MS estimated position is given by the coordinates associated with $\vec{F}_k$, obtained from the 4-uple $\left(\vec{F}_k, x, y, z\right)$ stored in the CDB.

III. DIAGRAM OF THE PROPOSED SOLUTION

Fig. 1 shows a diagram of the proposed solution, in the post-training phase or *on-line* phase. Initially, the **TFing** is transported to the principal components subspace, through principal components analysis (PCA). Then, it is presented to the classifier - K-medians or Kohonen layer - which identifies the cluster which the **TFing** belongs to. Then, in the N-dimensional RSS space, the **TFing** is compared to all the **RFing**'s within the selected cluster. Finally, the process returns the geographic coordinates (x, y, z) of the measurement point containing the **RFing** with the highest similarity with the **TFing**.

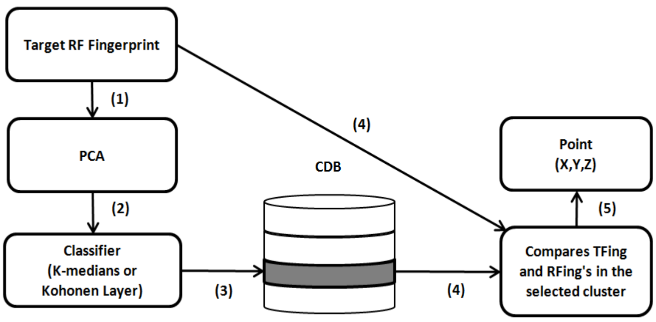

Fig. 1. Diagram of the Proposed Solution.

IV. UNSUPERVISED CLUSTERING

IV-A. Justification for the use of Unsupervised Training

Due to the inherent complexity of the RF channel, it is not possible to know beforehand how the **RFing**'s - and consequently the measurement points where they were collected - will cluster. Under certain propagation conditions, measurement points far away from each other in the Euclidean three-dimensional space might have RF fingerprints which are close together in the N-dimensional RSS space. Therefore, for instance, it is not possible to ascertain that measurement points in the same floor will belong to the same cluster. So, there are no predefined targets during the training phase, and it is up to the classifier to identify, without supervision, how the **RFing**'s will cluster together. Note that, as each **RFing**'s is georeferenced, i.e., is associated with a measurement point with known coordinates, by grouping the **RFing**'s in the N-dimensional RSS space, the classifier is indirectly grouping the measurement points in the Euclidean three-dimensional space.

IV-B. Identifying the Optimum Number of Clusters

In order to obtain a position fix without clustering or any other search space reduction technique, the **TFing** must be compared with all **RFing**'s stored in the CDB. With clustering, positioning is carried out in two phases: **i)** the **TFing** is presented to the trained classifier, which identifies the cluster the **TFing** belongs to; **ii)** the **TFing** is compared only to the **RFing**'s belonging to the selected cluster. In the first phase, the comparison is carried out in the subspace of the principal components. In the second phase, the comparison is carried out in the N-dimensional RSS space.

This scheme results in the reduction of the computational complexity of the DCM algorithm, considering that the computational complexity is numerically equal to the number of comparisons between RF fingerprints during a position fix. Without clustering, each **TFing** is compared to N_p **RFing**'s, where N_p is the number of RF fingerprints stored in the CDB. With clustering, and assuming clusters with the same number of elements - which is a reasonable approximation, if it is assumed that a Kohonen layer with conscience is being used [13] - each **TFing** is compared to, in the first phase, to N_c fingerprints, corresponding to the centers of each cluster. In the second phase, **TFing** is compared to the N_p/N_c **RFing**'s belonging to the cluster selected in the first phase. The computational complexity of the positioning method can then be expressed as a function of the number of clusters N_c as:

$$f\left(N_c\right) = N_c + N_p/N_c \tag{2}$$

The theoretical optimum value of N_c is that which minimizes the function f. By solving $\partial f/\partial N_c = 0$, one obtains:

$$N_c = \sqrt{N_p} \tag{3}$$

Depending on the distribution of the **RFing**'s in the input space, and on the clustering technique applied, it might not be possible to map a number of clusters equal to the theoretical optimum value of N_c during the unsupervised training.

IV-C. Unsupervised Clustering Techniques

Two unsupervised clustering techniques were used in this work: K-medians [14] and Kohonen layer (with and without conscience) [13].

K-Medians is a variation of K-means, which is a classic unsupervised clustering technique. Both follow the same steps: **i)** N_c clusters centers are defined; **ii)** for each input vector, the distance between the vector and each cluster center is calculated; **iii)** the input vector is associated with the cluster with the closest center. Steps **(i)** to **(iii)** are repeated for every input vector. At the end of the first epoch, i.e., after all input vectors have been presented to the classifier once, the clusters centers are re-calculated, as well as the distances between each input vector and the new clusters centers. If any vector switches clusters, the process is repeated until no vector switches clusters or until a maximum number of epochs is reached. The difference between K-means and K-medians lies on how the clusters centers are calculated. Whilst in K-means, the clusters centers are given by the barycenter of the vectors of each cluster, in K-medians, instead of the arithmetic mean, the median is used. The advantage of using the median is that it is a more robust estimator, less susceptible to outliers, i.e., input vectors with much noise [14].

The Kohonen layer used in this work is one-dimensional with a neighbor radius equal to zero, which means that only the winner neuron is activated. This classifier has an input layer and a competitive layer, as shown in Fig. 2. For each input vector $\vec{X} = [X_1\ X_2 \ldots X_N]$, the winner neuron at the competitive layer is that whose synapse $\vec{W}_i$ is the most similar to the input vector $\vec{X}$. The output of the winner neuron is activated (y_i=1), while the outputs of all other neurons remain equal to zero (y_j=0, $\forall j \neq i$) [15]. The similarity measure used was:

$$u_i = -\left(\vec{X} - \vec{W}_i\right)\left(\vec{X} - \vec{W}_i\right)^T - w_{i,0}, \forall i \in [1, 2, \ldots, N_c] \qquad (4)$$

where $w_{i,0}$ is the i-th neuron conscience bias, which is defined by Eq. (8).

Just like in K-means, the initialization of the synaptic weights is critical. The technique used for the initialization is described in Section V. In order to distributed the training approximately evenly among all neurons in the competitive layer, conscience is used. With conscience, neurons which are constantly winning receive a progressively decreasing negative *bias*. This allows less trained neurons to become more similar to the input vectors. With the conscience mechanism, neurons in the Kohonen layer naturally represent approximately equal amount of information [16]. Individual decreasing learning steps per neuron are also used [15].

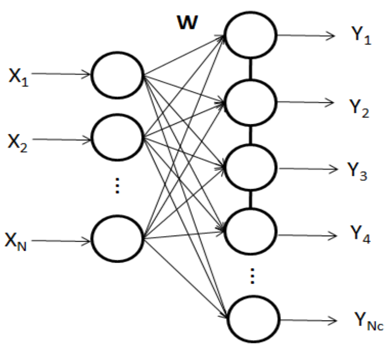

Fig. 2. ANN with one competitive layer.

V. EXPERIMENTAL EVALUATION

V-A. Experiment Setup

The WiFi RSS measurement campaign was carried out in the 12 floors of Principal Joao Lyra Filho Pavilion at the University of Rio de Janeiro State (UERJ) [17]. The software used to collect the WiFi scans was *NetStumbler* version 0.4, which was run in a Toshiba A75-S211 laptop with a Atheros AR5005GS built-in 802.11b/g adapter. *NetStumbler* forces the WiFi adapter to carry out a passive scan of 802.11 networks, i.e., without sending probe requests. During the passive scan, the WiFi adapter remains a certain time period on each channel, waiting to receive a beacon. The beacon, which is sent by every WiFi access point (AP), contains the network identifier (SSID - *Service Set ID*) and the AP MAC (*Medium Access Control*) address [18]. For each detected AP, *NetStumbler* stores the MAC, SSID, carrier number, noise level and signal-to-noise ratio. The laptop was placed over a wheeled table, and at each of the 924 measurement points the WiFi adapter collected between 180 and 240 WiFi scans, at a rate of one per second. Each WiFi scan contains data from several AP's. Fig. 3 shows a perspective spatial view of the measurement points positions in the UERJ main building.

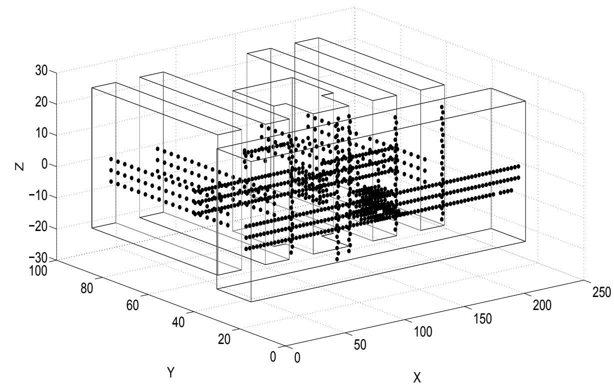

Fig. 3. Measurement points at UERJ building.

Each measurement point is identified by a unique **RFing** containing the mean RSS values of each detected WiFi network. At each measurement point, the mean RSS values per WiFi network are calculated using all WiFi scans carried out in that particular point. If a WiFi network is detected in less than 10% of the WiFi scans at a given point, its mean RSS value is assumed to be zero. A total of $N = 136$ WiFi networks were detected, resulting in a training set matrix of size 924×136.

The first 50 WiFi scans at each measurement point were used as input vectors of the test set. Considering all measurement points, a total of 924×50 WiFi scans was selected for the test set, resulting in a test set matrix of size 46200×136.

Prior to presenting the **RFing**'s to the classifier, the RSS values were converted to the logarithmic scale in order to compress them to the numerical range -120 to -30 dBm. Then, as shown in Fig. 1, PCA was applied to reduce the input vectors dimension. PCA generates a new set of mutually orthogonal variables called principal components (PC's). Firstly, the training set, with M input vectors, is translated by extracting the sample mean at each dimension, obtaining the matrix:

$$\mathbf{B} = \left[v_{i,j} - \bar{V}_j\right]_{i=1,\ldots,M;j=1,\ldots,N} \qquad (5)$$

where $v_{i,j}$ is the j-th WiFi network RSS value at the i-th scan, and $\bar{V}_j$ is j-th network RSS sample mean. Let $\vec{U}_j$ be the j-th eigenvector of the covariance matrix of $\mathbf{B}$. The PC's matrix of the training set is given by:

$$\mathbf{P}_{M\times N} = \mathbf{B}\cdot\left[\vec{U}_j\right]_{j=1,\ldots,N} \quad (6)$$

The columns in **P** are sorted in decreasing variance order, each one corresponding to a PC. To reduce the training patterns dimension, only the first p PC's are held. In this paper, p was selected so that at least 99% of the training set total variance was preserved, resulting in $p = 78$.

The sample means and the matrix $\left[\vec{U}_j\right]_{j=1,\ldots,N}$, obtained during application of PCA to the training set, are also used to project each test set vector into the PC's subspace. Just like in the training set, only the first p PC's are held.

The cluster central patterns, in the case of the K-medians, or the synaptic weights, in the case of the Kohonen layer, are defined in the PC's subspace. So, there is a $N_c \times p$ matrix of cluster central patterns, where N_c is the number of clusters and p is the number of PC's that are kept ($p = 78$). The theoretical optimum value of N_c is given by Eq. (3). For $N_p = 924$, one has $N_c = 30$.

For the initialization of the clusters central patterns (or the synaptic weights) in the PC's subspace, the maximum variance dimension (the first PC) has been divided into N_c equal length sections. In each section, a input vector whose first PC value is equal to the section median is selected. The selected input vectors are the initial values for the N_c clusters central patterns [19].

During the Kohonen layer training, individual decreasing learning steps per synapse were used, as defined by:

$$\alpha_i(n) = \alpha_0 \exp(-n/N_0) \quad (7)$$

where $\alpha_i(n)$ is the i-th synapse learning step at instant n, $\alpha_0 = 0.45$ e $N_0 = 120$.

Conscience was used in the training phase. Conscience is implemented by adding a negative bias to the neuron similarity function, which was defined by Eq. (4. The negative bias is given by:

$$w_{i,0} = \phi^2\left\{0.5\left[1-\tanh\left(k\left(p_i - p^*\right)\right)\right] - 1\right\} \quad (8)$$

where ϕ is equal to the diameter of the single class (i.e., assuming that all vectors belong to only one class and finding the highest distance between any pair of input vectors) in the PC's subspace, $k = 4.5$, p_i is the percentage of times the i-th neuron is trained (number of times the neuron was trained or won over the number of input vectors presented to the classifier) and $p^* = 1/N_c$.

V-B. Experiment Results

Fig. 4 shows the total internal dispersion per cluster ($F_{in,i}$; $i = 1,\ldots,N_c$), as defined by Eq. (9).

$$F_{in,i} = \sum_{j=1}^{n_i}\left\|\vec{X}_{i,j} - \vec{M}_i\right\|^2 \quad (9)$$

where n_i is the number of elements in the i-th cluster, $\vec{X}_{i,j}$ is the j-th element of the i-th cluster, and $\vec{M}_i$ is the arithmetic mean of the i-th cluster elements.

The total internal dispersion for all clusters, as defined by Eq. (10), is a parameter to be minimized during the classifier training. Among the three evaluated classifiers, the Kohonen layer with conscience is the one produces the lowest total internal dispersion ($F_{in} = 1.8334 \times 10^6$), followed by the Kohonen layer without conscience ($F_{in} = 2.1379 \times 10^6$) and by K-medians ($F_{in} = 2.7974 \times 10^6$).

$$F_{in} = \sum_{i=1}^{N_c} F_{in,i} \quad (10)$$

Another parameter used to evaluate the classifier performance is its precision, defined by the ratio TP / (TP + FP), where TP is the number of true positives (number of input vectors belonging to a given cluster and which were correctly identified as such) and FP is the number of false positives (number of input vectors not belonging to a given cluster and which were incorrectly identified as belonging to that cluster). Fig. 5 shows the precision per cluster per classifier. The Kohonen layer with conscience, even though not achieving the highest precision for all clusters, provides the best global result, reaching a precision higher then 85% in 29 out of the 30 clusters.

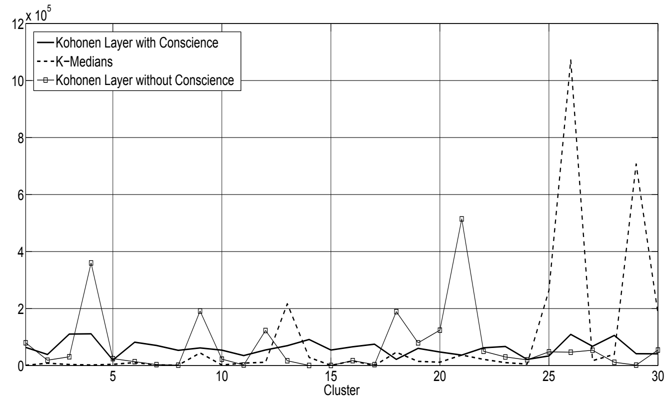

Fig. 4. Total Internal Dispersion per Cluster.

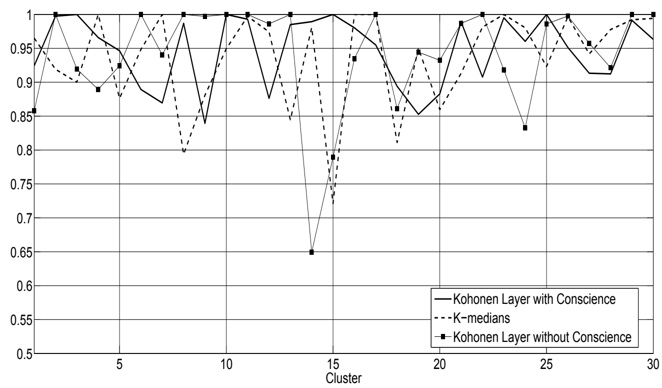

Fig. 5. Classification Precision per Cluster.

By limiting the search area to the cluster to which the **TFing** belongs, clustering is expected to reduce the DCM positioning error. This assumption is confirmed by the results in Table I, which shows the average, 75th and 90th percentiles positioning errors, as well as the floor identification accuracy. The use of Kohonen layer with conscience to reduce the search space within the CDB results in a 38% reduction in the average positioning error. The positioning errors for the 75th and 90th percentiles are reduced by 42% and 45%, respectively. The error in floor identification is reduced by 6%. This latter reduction is small, as measurement points within a cluster are not necessarily restricted to a single floor, as shown by Fig. 6.

Table I. DCM positioning error (average, 75th and 90th percentiles) in meters and floor identification accuracy.

Method	Avg Error	75th Percentile	90th Percentile	Floor Id. Accuracy
Without clustering	7.6	8.6	27.5	78%
K-medians	5.5	8.5	18.0	83%
Kohonen Layer	5.2	6.1	17.3	83%
Kohonen Layer with Conscience	4.7	6.0	15.0	84%

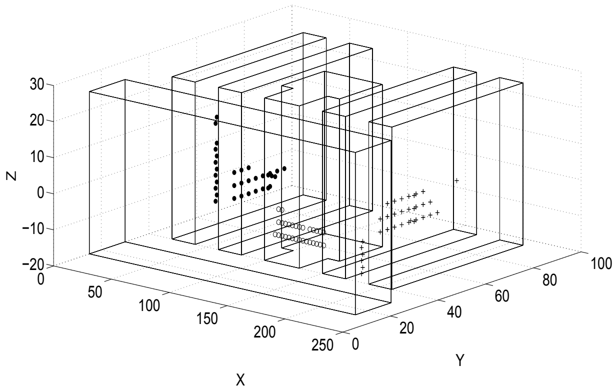

Fig. 6. Clusters 12, 17(+) e 20(o)

Using the definition of computational complexity presented in Section IV-B, it is possible to calculate a theoretical factor δ_{max}, which express the maximum reduction of computational complexity. This factor would be achieved if all clusters had the same number of elements, and is given by:

$$\delta_{max} = [N_p - f(N_c)]/N_p \tag{11}$$

where $f(N_c)$ is given by Eq. (2). For $N_p = 924$ and $N_c = 30$, $\delta_{max} = 93.4\%$. In practice, clusters do not have the same number of elements, and the computational complexity reduction is given by:

$$\delta = \frac{\left(M \cdot N_p + \sum_{i=1}^{M} n_i\right)}{M \cdot N_p} \tag{12}$$

where n_i is the number of elements within the cluster which the i-th input vector (i.e., the i-th **TFing**) was associated with, M is the number of **TFing**'s and $N_p = 924$. Note that δ will always be less than δ_{max}, as long as the number of **TFing**'s per measurement point is constant. The experimental value of δ was evaluated for the three classifiers: Kohonen layer with conscience (92.9%) and without conscience (87.3%), and K-medians (84.1%). The Kohonen layer with conscience is the classifier which most reduces the computational complexity of the DCM algorithm, almost reaching δ_{max}. It happens because conscience tends to make the neurons in the competitive layer map equal regions of the input space, resulting in clusters with approximately the same number of elements.

Fig. 7 shows the time to produce each position fix. Without clustering, the average time per position fix is 3.7 ms. Using the Kohonen layer with conscience, this value drops to 0.7 ms, a 81% reduction.

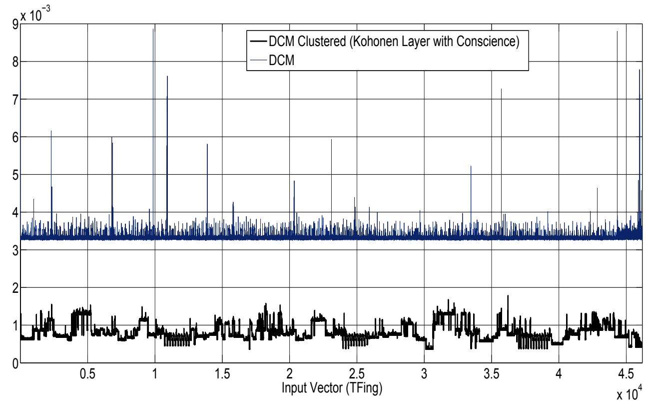

Fig. 7. Time to obtain each position fix (milliseconds).

V-C. Comparison with Published Results

The use of Kohonen layer with conscience has reduced the DCM algorithm computational complexity in approximately 93%. In [19], the proposed clustering scheme achieved a 91% computational complexity reduction in relation to the RADAR method [20]. However, as the relation between the computational complexity of RADAR and DCM has not been established, a direct comparison between the effects of the clustering techniques used in this work and in [19] is not straightforward.

In [20], the positioning error for the 75th percentile was 4.7 meters when the **TFing**'s were built using mean RSS values of 20 samples per measurement point. When just one sample is used per **TFing** (as in this work), the positioning error for this same percentile rises to 6.1 meters. This precision is approximately the same achieved when the Kohonen layer with conscience is used together with the DCM, as shown in Table I.

In [21], a GSM fingerprinting indoor localization system for multi-floor positioning was proposed, achieving a 73% floor identification accuracy. The Kohonen Layer with conscience + DCM scheme proposed in this work achieves a floor identification precision of 84%.

VI. CONCLUSION

In this work, the use of unsupervised clustering techniques to reduce the search space within the CDB in DCM positioning was analyzed. Using a Kohonen layer with conscience, the average time to produce a position fix was reduced 81%, and the average positioning error was reduced 38%, in relation to the DCM positioning without clustering. A 6% reduction in the floor identification error was also observed.

VII. REFERENCES

[1] Federal Communications Commission, “Guidelines for Testing and Verifying the Accuracy of Wireless E911 Location Systems,” *OET BULLETIN No.71*, 2000.

[2] Hui Liu, Houshang Darabi, Pat Banerjee, and Jing Liu, “Survey of Wireless Indoor Positioning Techniques and Systems,” *IEEE Transactions on Systems, Man and Cybernetics - Part C: Applications and Reviews*, vol. 37, no. 6, pp. 1067–1080, November 2007.

[3] K. Thomas et al, “On-demand fingerprint selection for 802.11-based positioning systems,” in *Proc. of 2008 IEEE international Symposium on a World of Wireless, Mobile and Multimedia Networks*, Newport Beach, USA, June 2008, pp. 1–8.

[4] H. Laitinen, J. Lahteenmaki, and T. Nordstrom, “Database Correlation Method for GSM Location,” in *Proceedings of IEEE 53rd Vehicular Technology Conference*, Rhodes, Greece, May 2001, pp. 2504–2508.

[5] R. S. Campos and L. Lovisolo, “RF Fingerprinting Location Techniques,” in *Handbook of Position Location: Theory, Practice, and Advances*, John Wiley and Sons, Ed., chapter 15, pp. 487–520. S. A. Zekavat and R. M. Buehrer, 1 edition, 2011.

[6] Sinan Gezici, “A Survey on Wireless Position Estimation,” *Wireless Personal Communications: An International Journal*, vol. 44, no. 3, pp. 263–282, February 2008.

[7] J. Zhu and G. D. Durgin, “Indoor/Outdoor Location of Cellular Handsets Based on Received Signal Strength,” *Eletronic Letters*, vol. 41, no. 1, pp. 24–26, January 2005.

[8] R. S. Campos and L. Lovisolo, “A Fast Database Correlation Algorithm for Localization of Wireless Network Mobile Nodes using Coverage Prediction and Round Trip Delay,” in *Proceedings of IEEE 69th Vehicular Technology Conference*, Barcelona, Spain, April 2009.

[9] M. J. Magro and C. J. Debono, “A Genetic Algorithm Approach to User Location Estimation in UMTS Networks,” in *EUROCON - The International Conference on Computer as a Tool*, Warsaw, Poland, September 2007, pp. 1136–1139.

[10] M. Hata, “Empirical Formula for Propagation Loss in Land Mobile Radio Services,” *IEEE Transactions on Vehicular Technology*, vol. 29, no. 3, pp. 317–325, August 1980.

[11] M. Youssef et al, “WLAN Location Determination via Clustering and Probability Distributions,” in *Proc. of the First IEEE International Conference on Pervasive Computing and Communications*, Seattle, USA, March 2003, pp. 143–150.

[12] M. A. Spirito, Mauricio Caceres, and Francesco Sottile, “WLAN-Based Real Time Vehicle Locating System,” in *Proceedings of IEEE 69th Vehicular Technology Conference*, Barcelona, Spain, April 2009.

[13] D. DeSieno, “Adding a conscience to competitive learning,” in *Proc. of 1988 IEEE International Conference on Neural Networks*, Sand Diego, USA, July 1988, pp. 117–124.

[14] S. Marsland, *Machine Learning - An Algorithmic Perspective*, CRC Press, 2009.

[15] Simon Haykin, *Neural Networks: A Comprehensive Foundation*, Prentice Hall PTR, 1994.

[16] R. G. Silva, “Improved Self-Organizing Map Performance Applied To Process Monitoring,” in *Proc. of 6 Congreso Interamericano de Computacion Aplicada a La Industria de Procesos*, Puebla, Mexico, October 2003.

[17] Adriano Souza Gonzaga, Alexandre Veiga Maral, and Ivan Klavin Ferreira, “Survey of WiFi 802.11 Networks at UERJ Campus,” 2008, Graduation Project - University of Rio de Janeiro State(UERJ).

[18] IEEE Computer Society, “IEEE Standard for Information Technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements; Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications,” 2007.

[19] M. Belal and A. Daoud, “A New Algorithm for Cluster Initialization,” in *Proc. of the Second World Enformatika Conference*, Istanbul, Turkey, 2005.

[20] P. Bahl and V. N. Padmanabhan, “RADAR: An In-Building RF-based User Location and Tracking System,” in *Proceedings of 19th Annual Joint Conference of the IEEE Computer and Communications Societies*, Tel Aviv, Israel, March 2000, pp. 775–784.

[21] A. Varshavsky et al, “The SkyLoc Floor Locatization System,” in *Proceedings of the Fifth Annual IEEE International Conference on Pervasive Computing and Communications (PerCom’07)*, White Plains, USA, March 2007, pp. 125–134.