DECENTRALIZED LOCALIZATION USING FINGERPRINTING AND KERNEL METHODS IN WIRELESS SENSOR NETWORKS

Sandy Mahfouz(1), *Farah Mourad-Chehade*(1), *Paul Honeine*(1), *Joumana Farah*(2), *Hichem Snoussi*(1)

(1)Institut Charles Delaunay (CNRS), Université de Technologie de Troyes, Troyes, France
(2)Department of telecommunications, Faculty of engineering, Université Saint-Esprit de Kaslik, Lebanon

ABSTRACT

Sensors localization has become an essential issue in wireless sensor networks. This paper presents a decentralized localization algorithm that makes use of radio-location fingerprinting and kernel methods. The proposed algorithm consists of dividing the network into several zones, each of which having a calculator capable of emitting, receiving and processing data. By using radio-information of its zone, each calculator constructs, by means of kernel methods, a model estimating the nodes positions. Calculators estimates are then combined together, leading to final position estimates. Compared to centralized methods, this technique is more robust, less energy-consuming, with a lower processing complexity.

Index Terms— decentralized processing, fingerprinting, localization, machine learning, wireless sensor networks

1. INTRODUCTION

Recent advances in electronics and wireless communication technologies have led to the emergence of wireless sensor networks (WSNs). A WSN consists of a large number of tiny autonomous sensors, each of which being capable of sensing, processing, and transmitting environmental information [1]. One fundamental issue in WSNs is the sensors localization, since all collected data are meaningless if they are not coupled with their correct locations. A trivial solution to this problem consists of equipping each sensor with a positioning device such as the Global Positioning System (GPS) [2]. However, since WSNs are expected to scale up to thousands of sensors, using GPS is impractical. A better solution is to perform relative positioning, where two types of sensors are considered: *anchors* and *nodes*. While anchors have known fixed locations, nodes have unknown positions and thus they are localized with respect to anchors.

Many anchor-based localization algorithms have been proposed in literature, especially those based on estimating nodes distances to anchors using the Received Signal Strength Indicator (RSSI) [3, 4, 5, 6, 7]. RSSI methods have become popular due to their low-power consumption and cost competitiveness. However, using exact RSSIs to estimate distances is not always reliable, due to additive noise, obstacles, etc. A more robust RSSI-based approach performs fingerprinting [8, 9]. This technique consists in setting a certain number of positions in the environment called *offline* positions, then building a radio map where each reference position is associated to the RSSI measurements received at this position. A node's position is then determined by processing the RSSI values collected by this node using the constructed radio map. The advantage of location fingerprinting technique is that it takes into account the stationary characteristics of the environment such as multipath propagation, wall attenuation, etc.

In terms of computational concept, algorithms in WSNs can be divided into centralized and decentralized algorithms [10, 11]. The centralized algorithms require the transmission of all measurements to a fusion center (e.g., a sink node) for processing, which often results in unnecessary wasteful energy costs and bandwidth consumption and thereby reduces the lifetime and utility of the network. In decentralized algorithms, local information exchanged between neighboring nodes is processed locally and, hence, information processing is no more limited to one fusion center. Compared to the centralized strategy, the decentralized approach is more robust to failures, since several fusion centers are engaged. It is also less energy consuming and thus more adapted to WSNs limitations. In our previous work [12], we proposed a centralized localization algorithm combining location fingerprinting and kernel methods. The processing of the collected data and all positions estimations were done by one single entity, i.e., a fusion center.

In this paper, we present a novel decentralized localization algorithm based on our previous work. The proposed algorithm consists in dividing the region of interest into distinct zones depending on the characteristics of the environment, and then setting for each zone a *calculator* performing computations. The localization algorithm is then run at each calculator, using an *offline* an *online* phase. In the *offline* phase, reference positions are generated in each zone, and RSSI measurements are collected at these positions, leading to a local power map representing the radio of each zone. Then, using kernel methods [13], each calculator defines a local model that associates to the RSSI measurements of its

This work is supported by the Champagne-Ardenne Region in France.

local map the geographical positions where they were collected. In the *online* phase, each node regularly collects RSSI measurements. Calculators use these RSSIs along with their models to compute local estimates of nodes positions. The local estimates are then combined, leading to one final estimate for each node at a given time. Having multiple calculators for performing estimations highly improves the robustness of the method.The rest of the paper is organized as follows. The proposed method is described in Section 2. In Section 3, the kernel methods are presented. Section 4 illustrates the effectiveness of the method. Finally, Section 5 concludes the paper.

2. DESCRIPTION OF THE PROPOSED METHOD

The objective of the proposed method is to compute the geographical locations of sensor nodes in an environment of D dimensions, with $D = 2$ for a 2D environment or $D = 3$ for a 3D environment. To this aim, two types of sensors are considered: anchors and mobile nodes. Anchors have known fixed locations, denoted by $\boldsymbol{a}_i,\ i \in \{1, \ldots, N_a\}$, and mobile nodes are moving with unknown positions, denoted by $\boldsymbol{x}_j(t),\ j \in \{1, ..., N_x\}$, and hence need to be regularly localized. The region of interest is then partitioned into Z distinct zones depending on the characteristics of the environment. Each zone has a calculator, that is a smart device (e.g., a small computer) capable of handling data, performing calculations, and exchanging information with the sensors. Calculators could also be anchors having high computation capabilities. These smart devices have fixed known locations and can be placed anywhere in a zone. Without loss of generality, we consider that zones are rectangular, each having one calculator located at their centers, as shown in Fig. 1. The localization is then performed locally at every calculator.

Each local algorithm consists of two *offline* and *online* phases. In the *offline* phase, $n^{(z)}$ *offline* positions, denoted by $\boldsymbol{p}_\ell^{(z)}$, $\ell \in 1, \ldots, n^{(z)}$, are generated in a uniformly distributed manner in the studied zone z, $z \in \{1, ..., Z\}$. Then, anchors broadcast signals in the network with the same initial power. Meanwhile, a sensor is temporarily placed at each *offline* position of each zone, it detects the anchors signals and measures their Received Signal Strength Indicators (RSSIs). All anchors signals are assumed to be received at all *offline* positions regardless of their zones. Let $\boldsymbol{\rho}_\ell^{(z)} = (\rho_{\boldsymbol{a}_1, \boldsymbol{p}_\ell^{(z)}}, \ldots, \rho_{\boldsymbol{a}_{N_a}, \boldsymbol{p}_\ell^{(z)}})^\top$ be the RSSI vector of signals received from the N_a anchors at the position $\boldsymbol{p}_\ell^{(z)}$, with $\ell \in \{1, ..., n^{(z)}\}$. Therefore, for each zone z, a local power map is obtained, composed of $n^{(z)}$ couples $\left(\boldsymbol{\rho}_\ell^{(z)}, \boldsymbol{p}_\ell^{(z)}\right)$. Once the local map is set, the algorithm aims at defining a function $\boldsymbol{\psi}^{(z)}(\cdot)$ that associates to each RSSI vector $\boldsymbol{\rho}_\ell^{(z)}$, for $\ell \in \{1, ..., n^{(z)}\}$, the corresponding position $\boldsymbol{p}_\ell^{(z)}$. Kernel methods provide an elegant framework to determine these functions, as it will be shown in Section 3.

In the *online* phase, each node travels freely in the network and collects meanwhile RSSIs from anchors. Let $\boldsymbol{\rho}_j(t)$ be the N_a-vector of RSSIs collected by the node j, at time t, from the N_a anchors, with $j \in \{1, ..., N_x\}$. Then, each node sends its RSSI vector $\boldsymbol{\rho}_j(t)$ to all the calculators located within its communication range. Let $I_j(t)$ be the set of indices of the calculators detected by the node j at time t. Calculators referred in $I_j(t)$ apply their defined functions $\boldsymbol{\psi}^{(z)}(\cdot)$ to the RSSI vector $\boldsymbol{\rho}_j(t)$ to compute local estimates of the node's position as follows:

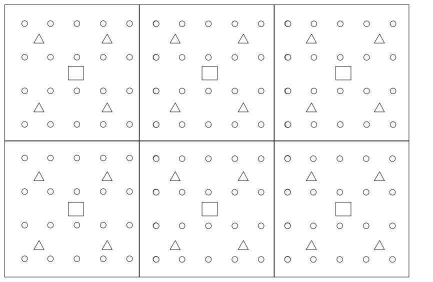

Fig. 1. Network topology- □ represents the calculators, △ represents the anchors, and ○ represents the *offline* positions.

$$\tilde{\boldsymbol{x}}_j^{(z)}(t) = \boldsymbol{\psi}^{(z)}(\boldsymbol{\rho}_j(t)),\ z \in I_j(t),\ j \in \{1, ..., N_x\}.$$

Then, local estimates are sent to the nearest calculator, where a global estimate is given by a combination of all local estimates using:

$$\tilde{\boldsymbol{x}}_j(t) = \sum_{z \in I_j(t)} w_z\, \tilde{\boldsymbol{x}}_j^{(z)}(t),\ j \in \{1, ..., N_x\}.$$

The quantities w_z are weights computed with respect to the distances separating the node from the calculators. Indeed, the more the node is close to a given calculator, the more the calculator's estimate is reliable and thus the greater the corresponding weight should be. Since distances to calculators are not available, we will use instead the powers of the messages emitted by the nodes to the calculators while sending their RSSI-vectors. Let $\xi_{z,j}(t)$ be the RSSI of the message sent by the node j to the calculator z at time t. Since signal powers decrease with the increase of their traveled distance, the closer the node j is to the calculator z, the greater $\xi_{z,j}(t)$ theoretically is. Then, the weights w_z are given by $\frac{\xi_{z,j}(t)}{\sum_{v \in I_j(t)} \xi_{v,j}(t)}$, $z \in I_j(t)$. By using several computation points, the proposed method is more robust to failures compared to the centralized scheme where only one fusion center is considered. Indeed, a failure of the fusion center in the centralized method is fatal, whereas it is much less binding in our decentralized method, since many fusion centers, that is calculators, exist. Also, the localization in this method is less energy consuming, since nodes only send information to calculators in their sensing range, whereas in the centralized scheme data must be routed to a single fusion center regardless of the distance to be traveled.

3. ESTIMATION OF THE LOCAL MODELS

In this section, we take advantage of kernel-based machine learning to define, for each zone z, a model $\boldsymbol{\psi}^{(z)}(\cdot)$ that associates to each RSSI vector $\boldsymbol{\rho}_\ell^{(z)}$, for $\ell \in \{1, ..., n^{(z)}\}$, the corresponding position $\boldsymbol{p}_\ell^{(z)}$. To this end, let $\boldsymbol{\psi}^{(z)}(\cdot) = \big(\psi_1^{(z)}(\cdot), \ldots, \psi_D^{(z)}(\cdot)\big)^\top$, where $\psi_d^{(z)} : \mathbb{R}^{N_a} \mapsto \mathbb{R}$ estimates the d-th coordinate. Thus, the function $\psi_d^{(z)}$, for an input $\boldsymbol{\rho}_\ell^{(z)}$, yields the d-th coordinate in $\boldsymbol{p}_\ell^{(z)} = (p_{\ell,1}^{(z)}, \ldots, p_{\ell,D}^{(z)})^\top$. Let us consider a reproducing kernel $\kappa : \mathbb{R}^{N_a} \times \mathbb{R}^{N_a} \mapsto \mathbb{R}$, and let $\mathcal{H}$ be the induced space.

The function $\psi_d^{(z)} \in \mathcal{H}$ is estimated by minimizing the mean quadratic error between the model's outputs $\psi_d^{(z)}(\boldsymbol{\rho}_\ell^{(z)})$ and the desired outputs $p_{\ell,d}^{(z)}$, for $\ell \in \{1, ..., n^{(z)}\}$:

$$\min_{\psi_d^{(z)} \in \mathcal{H}} \sum_{\ell=1}^{n^{(z)}} \big((p_{\ell,d}^{(z)} - \psi_d^{(z)}(\boldsymbol{\rho}_\ell^{(z)})\big)^2 + \eta^{(z)} \|\psi_d^{(z)}\|_{\mathcal{H}}^2, \qquad (1)$$

where the regularization parameter $\eta^{(z)}$ controls the tradeoff between the training error and the complexity of the solution. According to the representer theorem [14], the optimal function can be written as follows :

$$\psi_d^{(z)}(\cdot) = \sum_{\ell=1}^{n^{(z)}} \alpha_{\ell,d}^{(z)} \, \kappa(\boldsymbol{\rho}_\ell^{(z)}, \cdot). \qquad (2)$$

By injecting (2) in (1), we get the dual optimization problem:

$$\min_{\boldsymbol{\alpha}_d^{(z)}} \|\boldsymbol{p}_d^{(z)} - \boldsymbol{K}^{(z)} \boldsymbol{\alpha}_d^{(z)}\|^2 + \eta^{(z)} (\boldsymbol{\alpha}_d^{(z)})^\top \boldsymbol{K}^{(z)} \boldsymbol{\alpha}_d^{(z)},$$

where $\boldsymbol{\alpha}_d^{(z)} = \big(\alpha_{1,d}^{(z)}, \ldots, \alpha_{n^{(z)},d}^{(z)}\big)^\top$, $\boldsymbol{p}_d^{(z)}$ is the vector of entries $p_{\ell,d}^{(z)}$ for $\ell \in \{1, ..., n^{(z)}\}$, and $\boldsymbol{K}^{(z)}$ is a $n^{(z)}$-by-$n^{(z)}$ matrix whose (s, s')-th entry is given by $\kappa(\boldsymbol{\rho}_s^{(z)}, \boldsymbol{\rho}_{s'}^{(z)})$, for $s, s' \in \{1, ..., n^{(z)}\}$. The solution of the above problem is given by the following linear system :

$$(\boldsymbol{K}^{(z)} + \eta^{(z)} \boldsymbol{I}_{n^{(z)}}) \, \boldsymbol{\alpha}_d^{(z)} = \boldsymbol{p}_d^{(z)}, \qquad (3)$$

where $\boldsymbol{I}_{n^{(z)}}$ is the $n^{(z)}$-by-$n^{(z)}$ identity matrix. It is easy to see the impact of the regularization parameter on the well-posedness of the problem, since the matrix between parenthesis is always non-singular for an appropriate value of $\eta^{(z)}$.

Equation (3) shows that the same matrix $(\boldsymbol{K}^{(z)} + \eta^{(z)} \boldsymbol{I}_{n^{(z)}})$ needs to be inverted in order to estimate each coordinate d. Nevertheless, it is reasonable to collect all D estimations (D being the space's dimension) in a single matrix inversion problem, thus reducing the computational complexity:

$$\boldsymbol{\alpha}^{(z)} = (\boldsymbol{K}^{(z)} + \eta^{(z)} \boldsymbol{I}_{n^{(z)}})^{-1} \boldsymbol{P}^{(z)}, \qquad (4)$$

where $\boldsymbol{\alpha}^{(z)} = \big(\boldsymbol{\alpha}_1^{(z)}, \ldots, \boldsymbol{\alpha}_D^{(z)}\big)$ and $\boldsymbol{P}^{(z)} = \big(\boldsymbol{p}_1^{(z)}, \ldots, \boldsymbol{p}_D^{(z)}\big)$. Using equation (2) and the definition of the vector of functions $\boldsymbol{\psi}^{(z)}(\cdot)$, we define now a model for each zone that allows us to find the D coordinates at once as follows:

$$\boldsymbol{\psi}^{(z)}(\cdot) = \sum_{\ell=1}^{n^{(z)}} \boldsymbol{\alpha}^{(z)} \kappa(\boldsymbol{\rho}_\ell^{(z)}, \cdot).$$

We have now defined, for each zone z, a model $\boldsymbol{\psi}^{(z)}$ that associates to each RSSI vector $\boldsymbol{\rho}_\ell^{(z)}$, for $\ell \in \{1, ..., n^{(z)}\}$, the corresponding position $\boldsymbol{p}_\ell^{(z)}$. The Z models are used for localization as explained at the end of Section 2.

In this paper, we consider the Gaussian kernel given by:

$$\kappa(\boldsymbol{\rho}_s, \boldsymbol{\rho}_{s'}) = \exp\left(\frac{-\|\boldsymbol{\rho}_s - \boldsymbol{\rho}_{s'}\|^2}{2(\sigma^{(z)})^2}\right),$$

where $\|\cdot\|$ is the Euclidian norm and $\sigma^{(z)}$ the bandwidth of the Gaussian kernel associated to zone/calculator z. This quantity, together with the regularization parameter $\eta^{(z)}$, control the degree of smoothness, noise tolerance, and generalization of the solution. Since we have different data and a different model for each zone, $\sigma^{(z)}$ and $\eta^{(z)}$ vary from zone to zone. Choosing $\sigma^{(z)}$ and $\eta^{(z)}$ for each zone is performed using the cross-validation technique. This approach is a statistical method of evaluating and comparing learning algorithms by dividing data into two segments: one for training the model and the other one for validating it. We employ this technique to choose proper values for $\eta^{(z)}$ and $\sigma^{(z)}$ for each model z [15]. In this paper, we use the k-fold cross-validation which is the basic form of cross-validation. The data in each zone z is first partitioned into k roughly equal sized folds. Subsequently, k iterations of training and validation are performed such that, within each iteration, $k-1$ folds are used for learning and the remaining one for validation. In each iteration, we estimate the error on the validation set for different values of $\eta^{(z)}$ and $\sigma^{(z)}$. In the end, we choose $\eta^{(z)}$ and $\sigma^{(z)}$ that give a minimum error for all iterations.

4. SIMULATIONS

In this section, we evaluate the performance of the proposed method using various scenarios. In the first part, the proposed method is tested with Matlab-generated data. In the second section, the results are shown in the case of real data gathered in a $10m \times 10m$ real indoor environment [16]. In the final section, the results obtained with the proposed method are compared to the ones obtained when performing localization using connectivity information or the Weighted K-Nearest Neighbor (WKNN) algorithm.

4.1. Evaluation of the proposed method on Matlab-simulated data

We consider a $120m \times 120m$ area partitioned into four equal zones and we generate 16 static anchors uniformly distributed over the area. For the training phase, 144 *offline* positions are generated in a uniformly distributed manner over the area, leading to $n^{(z)} = 36$ *offline* positions per zone, $z = 1, ..., 4$. The RSSI values needed to construct the local power maps are generated using the well-known Okumura-Hata model [17] given by:

$$\rho_{\boldsymbol{a}_i, \boldsymbol{p}_\ell^{(z)}} = \rho_0 - 10\, n_P \, \log_{10} \|\boldsymbol{a}_i - \boldsymbol{p}_\ell^{(z)}\| + \varepsilon_{i,\ell}^{(z)},$$

where $\rho_{\boldsymbol{a}_i, \boldsymbol{p}_\ell^{(z)}}$ (in dBm) is the power received from the anchor $\boldsymbol{a}_i$ by the node at position $\boldsymbol{p}_\ell^{(z)}$, ρ_0 is the initial power (in dBm) set to 150, n_P is the path-loss exponent set to 4, $\|\boldsymbol{a}_i - \boldsymbol{p}_\ell^{(z)}\|$ is the Euclidian distance between the position $\boldsymbol{p}_\ell^{(z)}$ of the considered node and the anchor position $\boldsymbol{a}_i$, and $\varepsilon_{i,\ell}^{(z)}$ is the noise affecting the RSS measures. We use cross-validation to find the proper values of the tunable parameters $\eta^{(z)}$ and $\sigma^{(z)}$ for each model z, considering $\eta^{(z)} = 2^s$ with $s \in \{-20, -19, \cdots, -1\}$ and $\sigma^{(z)} = 2^{s'}$ with $s' \in \{1, 2, \cdots, 10\}$.

We then consider the localization problem of one mobile node. To this end, we generate a trajectory and calculate the RSSI values collected by the moving node using the same RSSI model as for the *offline* positions. The powers of messages exchanged between the node and the calculators are also computed using this model and the node is assumed to communicate with all calculators. Fig. 2 shows the generated trajectory and the estimated one using noiseless RSSIs. It is obvious that the estimation is almost exact. The estimation error, measured by the root mean squared distance between the exact positions and the estimated ones, is in this case equal to $0.14m$. If we consider the centralized version of this method [12], the error is equal to $0.16m$. To evaluate the robustness of our method against noise, we add a zero-mean Gaussian white noise of standard deviation $\sigma_b = 1dB$ to the RSSI values. The estimated trajectory still follows the actual one and the estimation error is equal to $1.30m$, while the centralized version yields $1.23m$ of errors. Both centralized and decentralized methods have almost the same efficiency when the noise is spread in the same way all over the network. However, besides the robustness of the method, the decentralized method outperforms the centralized one in terms of computation complexity. Indeed, while our proposed method requires only $\mathcal{O}((n^{(z)})^3)$ for the matrix inversion, with $n^{(z)} = 36$ in this case, the centralized method considers all the *offline* positions at once with $\mathcal{O}(n^3)$, where $n = 144$.

4.2. Evaluation of the proposed method using real data

In this section, we study the performance of the new method in the case of real collected data. The set of collected measurements used are available from [16]. The authors deployed 48 uniformly distributed EyesIFX nodes in a room measuring approximately $10m \times 10m$. Furniture and people in the room cause multi-path interferences affecting the collected RSSI values. We chose 5 nodes to be anchors in our simulation and used the available data to generate 57 new *offline* positions in order to get a total of 100 *offline* positions. We partitioned the area into 9 zones and generated one calculator for each zone. To evaluate our algorithm in the case of a moving node, we generated a trajectory using the available data. Fig. 3 shows the estimated trajectories obtained using the proposed method and the centralized one. The estimation errors are equal to $0.28m$ and $0.37m$ respectively. By dividing the region into several zones, noises due to the space configuration being higher in some zones than in others would only affect some local models, the others leading to good estimates. This diversity is absent in the centralized model where anywhere noises impact the one and only model.

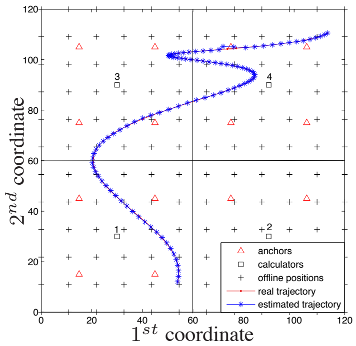

Fig. 2. Localization in the case of noiseless simulated data.

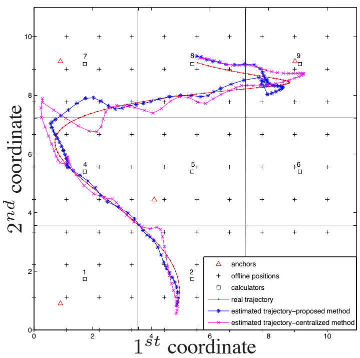

Fig. 3. Localization in the case of real data.

4.3. Comparison with previous techniques

In this section, we compare the results obtained when using the new method to the results obtained when using two different approaches: localization by connectivity and localization using the WKNN algorithm. We first consider the localization by connectivity [18] where nodes are localized using only neighboring anchors information. Assume that the transmission range is an ideal disk with equal transmission radius for all anchors. The position of an unknown node detecting its neighboring anchors is given by the intersection of all

Table 1. Comparison of the proposed method to the connectivity-based one.

No. of anchors N_a	16	49	81	100
Connectivity	8.43	4.07	3.68	3.25
Proposed method	0.14	0.20	1.04	1.37

Table 2. Comparison of the proposed method to the WKNN-based one, for different weight models.

"A" K=2	"B" K=4	"C" K=4	"D" K=10	"E" K=10	Proposed method
0.69	0.63	0.58	0.57	0.60	0.28

the range disks of these detected anchors. Table 1 shows the estimation error (in meters) in the case of noiseless Matlab-simulated data for different numbers of anchors. By adding anchors, the learning process in the proposed method leads to a rougher model with less freedom degrees, which yields increasing estimation errors. However, the error remains less than the connectivity-based one. We consider at second the WKNN algorithm [19] using fingerprinting coupled to RSSI-Euclidean distances computations. In WKNN, estimated positions are given by weighted combinations of the positions of the K nearest *offline* positions. Different weights are used in simulations of WKNN, as listed in Table III of [12], and the optimal value of K is determined using the cross-validation approach. Table 2 shows the estimation errors (in meters) obtained while estimating the same trajectory as in the case of real data (Fig. 3) with different weights of WKNN and with our method. The simulation results show that our method is more accurate than the WKNN based one. Note that localization by connectivity is not tested here because there are only 5 anchors in the network, which is insufficient for the connectivity technique to operate properly, while our approach is expected to yield good results in such case.

5. CONCLUSION

In this paper, we propose a novel decentralized localization method using radio-location fingerprinting and kernel methods. The proposed algorithm divides the network into several zones where different local position estimates are computed. A global estimate is then obtained by combining the local ones. Being a decentralized scheme, this method outperforms centralized ones in terms of robustness and computation complexity. Simulation results show that this method is more efficient than the WKNN algorithm for different weight models. It also gives better estimates than localization by connectivity for different number of anchors. Future works will handle improvements of this method, especially in terms of zones definitions and anchors signals ranges.

6. REFERENCES

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102 – 114, 2002, survey.

[2] B. Hofmann-Wellenhof, H. Lichtenegger, and J. Collins, *Global Positioning System: Theory and practice*. Springer-Verlag, 1994.

[3] R. Parker and S. Valaee, "Vehicular node localization using Received-Signal-Strength Indicator," *Vehicular Technology, IEEE Transactions on*, vol. 56, no. 6, pp. 3371 –3380, nov. 2007.

[4] P. Honeine, C. Richard, M. Essoloh, and H. Snoussi, "Localization in sensor networks - a matrix regression approach," in *Proc. IEEE Sensor Array and Multichannel Signal Processing Workshop*, Germany, 2008.

[5] M. Essoloh, C. Richard, H. Snoussi, and P. Honeine, "Distributed localization in wireless sensor networks as a pre-image problem in a reproducing kernel hilbert space," in *Proc. 16th European Conference on Signal Processing*, Lausanne, Switzerland, August 2008.

[6] F. Mourad, H. Chehade, H. Snoussi, L. Amodeo, and C. Richard, "Controlled mobility sensor networks for target tracking using ant colony optimization," *IEEE Transactions on Mobile Computing*, vol. 11, no. 8, pp. 1261 – 1273, 2012.

[7] F. Mourad-Chehade, P. Honeine, and H. Snoussi, "Polar interval-based localization in mobile sensor networks," *IEEE Transactions on Aerospace and Electronic Systems*, (in press) 2013.

[8] T.-N. Lin and P.-C. Lin, "Performance comparison of indoor positioning techniques based on location fingerprinting in wireless networks," in *Wireless Networks, Communications and Mobile Computing, International Conference on*, vol. 2, 2005, pp. 1569 – 1574 vol.2.

[9] J. Robles, M. Deicke, and R. Lehnert, "3D fingerprint-based localization for wireless sensor networks," in *Positioning Navigation and Communication (WPNC), 7th Workshop on*, march 2010, pp. 77 –85.

[10] E. F. Nakamura, A. A. F. Loureiro, and A. C. Frery, "Information fusion for wireless sensor networks: Methods, models, and classifications," *ACM Computing Surveys*, vol. 39, no. 3, August 2007.

[11] J.-P. Sheu, P.-C. Chen, and C.-S. Hsu, "A distributed localization scheme for wireless sensor networks with improved grid-scan and vector-based refinement," *Mobile Computing, IEEE Transactions on*, vol. 7, no. 9, pp. 1110–1123, Sept.

[12] S. Mahfouz, F. Mourad-Chehade, P. Honeine, J. Farah, and H. Snoussi, "Kernel-based localization using fingerprinting in wireless sensor networks," *14th IEEE International Workshop on Signal Processing Advances for Wireless Communications-SPAWC 2013, Submitted*, 2013.

[13] N. Aronszajn, "Theory of reproducing kernels," *Trans. Amer. Math. Soc.*, vol. 68, no. 3, pp. 337 – 404, 1950.

[14] B. Schölkopf, R. Herbrich, and A. J. Smola, "A generalized representer theorem," in *Proc. of the 14th Annual Conference on Computational Learning Theory and 5th European Conference on Computational Learning Theory*. London, UK, UK: Springer-Verlag, 2001, pp. 416–426.

[15] M. Stone, "Cross-validatory choice and assessment of statistical predictions," *J. Royal Stat. Soc.*, vol. 36, no. 2, pp. 111 – 147, 1974.

[16] G. Zanca, F. Zorzi, and A. Zanella, "RSSI measurements in indoor wireless sensor networks." [Online]. Available: http://telecom.dei.unipd.it/pages/read/59/

[17] A. Medeisis and A. Kajackas, "On the use of the universal Okumura-Hata propagation prediction model in rural areas," *Vehicular Conference Proceedings*, vol. 3, 2000.

[18] Y. Z. Y. Shang, W. Ruml and M. P. J. Fromherz, "Localization from mere connectivity," *MobiHoc 03: Proceedings of the 4th ACM international symposium on Mobile ad hoc networking and computing*, pp. 201 – 212, 2003.

[19] H. Koyuncu and S. H. Yang, "A 2D positioning system using WSNs in indoor environment," *International Journal of Electrical and Computer Sciences IJECS-IJENS*, vol. 11, no. 3, 2011.