

# Kernel-based localization using fingerprinting in wireless sensor networks

Sandy Mahfouz, Farah Mourad-Chehade, Paul Honeine, Hichem Snoussi  
Institut Charles Delaunay (CNRS)  
Université de Technologie de Troyes  
12, rue Marie Curie, 10010, France  
{sandy.mahfouz, farah.mourad, paul.honeine, hichem.snoussi}@utt.fr

Joumana Farah  
Department of telecommunications  
Faculty of engineering  
Université Saint-Esprit de Kaslik, Lebanon  
joumanafarah@usek.edu.lb

**Abstract**—Indoor localization is an important issue in wireless sensor networks for a very large number of applications. Recently, localization techniques based on the received signal strength indicator (RSSI) have been widely used due to their simple and low cost implementation. In this paper, we propose an algorithm for localization in wireless sensor networks based on radio-location fingerprinting and kernel methods. The proposed method is compared to another well-known localization algorithm in the case of real data collected in an indoor environment where RSSI measures are affected by noise and other interferences.

**Keywords**—fingerprinting, kernel methods, localization, wireless sensor networks

## I. INTRODUCTION

Wireless sensor networks (WSNs) have become a major research field during the last years. They are composed of a large number of tiny autonomous wireless sensors that are spatially distributed and can communicate and exchange data [1],[2]. The sensors are mostly battery powered; therefore, they have a limited lifetime, and hence all built-in algorithms must focus on reducing energy consumption. Nowadays, WSNs are rapidly gaining importance in many fields of applications, especially in military, environmental, and healthcare domains [3]. Indeed, WSNs can be efficiently used for monitoring or tracking of enemies in a battlefield [4], and for real-time forest fire detection [5]. Sensor networks can also be used to detect foreign chemical agents in the environment [6].

In all applications, awareness of location information is fundamental since collected data are meaningless without any geographical context. A direct way to locate the nodes is the use of GPS (Global Position System) devices; however, this solution is not practical because of the high cost of integrating GPS receivers and also of the limited efficiency of GPS signals in indoor environments. An alternative solution consists of considering two types of sensors, *anchors* and *non-anchor nodes*. Anchors have known fixed locations, whereas non-anchors nodes, or simply *nodes*, have unknown positions and thus need to be localized. Many localization algorithms using anchors have been proposed. They are mainly based on estimating the distances between the anchors and the nodes, using the Time Of Arrival (TOA) [8], the Time Difference Of Arrival (TDoA) [9], the Angle of Arrival (AoA) [10], or the Received Signal Strength Indicator (RSSI) [11],[12]. The RSSI methods are based on the attenuation of the signal strength over distance. They have become popular due to their low-power

consumption and cost competitiveness. For instance, in [13] and [14], interval-based methods using RSSI measurements are proposed; these methods perform an outer approximation of the solution area, leading to boxes guaranteed to include the actual location. In [15] and [16], the authors introduce a distributed strategy for localization in wireless sensor networks by solving the pre-image problem in a reproducing kernel Hilbert space.

Another technique for RSSI-based localization is fingerprinting [17],[18],[19]. The advantage of location fingerprinting technique is that it takes into account the stationary characteristics of the environment such as multipath propagation, i.e. wall attenuation. This technique consists of two phases: the *offline* phase and the *online* phase. The *offline* phase is a training phase where the goal is to build a training database for each reference location; a radio map is then created where each reference node has RSSI measures received from the anchors associated to its position. To determine the position of a node in the *online* phase, RSSI values collected from the node are processed and the estimated location coordinates are calculated based on the knowledge built during the *offline* phase. A well-known algorithm based on location fingerprinting is the Weighted K-Nearest Neighbor (WKNN) algorithm [20]. In the basic form of this algorithm, the RSSI values collected by the node whose position is unknown are compared to the samples in the database and the output is the weighted average of the  $K$  nearest samples; the nearness indicator for this method is based on the Euclidean distance between RSSIs.

In this paper, we propose a localization technique combining location fingerprinting and kernel methods. The proposed method employs fingerprinting information to define a model that associates to the measured data the positions where they are made. To this end, a fingerprinting database containing RSSI measures for each reference node associated with its position is constructed. The model is then defined using kernel methods ([21],[22]) and the constructed database. Then, in order to localize the nodes, each one collects, while moving, RSSI measures at every time-step. These RSSI measures are used with the defined model to estimate the position of the considered node. The rest of the paper is organized as follows. Section II outlines the proposed localization algorithm. Section III describes the use of kernel methods in our algorithm. Section IV illustrates the effectiveness of the method by comparing it to KNN-based methods. Finally, Section V concludes the paper.

## II. DESCRIPTION OF THE PROPOSED ALGORITHM

We consider two types of sensors: anchors and non-anchor nodes. Anchors have known fixed locations, denoted by  $\mathbf{a}_i$ ,  $i \in \{1, \dots, N_a\}$ , where  $N_a$  is their number. Non-anchor nodes or simply nodes are mobile having unknown positions, and hence needing to be localized regularly. Nodes positions at a given time  $t$  are denoted by  $\mathbf{x}_j(t)$ ,  $j \in \{1, \dots, N_x\}$ , where  $N_x$  is their number.

The proposed algorithm consists of two phases: *offline* and *online* phases. In the *offline* phase,  $N$  *offline* positions, denoted by  $\mathbf{p}_\ell$ ,  $\ell \in 1, \dots, N$ , are generated in a uniformly distributed manner in the studied environment. A node is then placed at each of these *offline* positions to perform measurements. Indeed, each anchor broadcasts signals in the network with the initial power  $\rho_0$ . By placing the node at a position  $\mathbf{p}_\ell$ , it becomes possible to measure the Received Signal Strength Indicator (RSSI) of anchors signals at this location. The RSSI is assumed to follow the Okumura-Hata model [23] given by:

$$\rho_{\mathbf{a}_i, \mathbf{p}_\ell} = \rho_0 - 10n_P \log_{10} \|\mathbf{a}_i - \mathbf{p}_\ell\| + \varepsilon_{i,\ell} \quad (1)$$

In this expression,  $\rho_{\mathbf{a}_i, \mathbf{p}_\ell}$  (in *dBm*) is the power received from the anchor  $\mathbf{a}_i$  by the node at position  $\mathbf{p}_\ell$ ,  $\rho_0$  is the initial power (in *dBm*),  $n_P$  is the path-loss exponent,  $\|\mathbf{a}_i - \mathbf{p}_\ell\|$  is the Euclidian distance between the considered node at position  $\mathbf{p}_\ell$  and the anchor  $\mathbf{a}_i$ , and  $\varepsilon_{i,\ell}$  is the noise affecting the RSS measures. Therefore, a database of  $n$  couples  $(\rho_\ell, \mathbf{p}_\ell)$  is obtained, where  $\ell \in \{1, \dots, N\}$ , and where  $\rho_\ell = (\rho_{\mathbf{a}_1, \mathbf{p}_\ell}, \dots, \rho_{\mathbf{a}_{N_a}, \mathbf{p}_\ell})^\top$  is the vector of received powers at position  $\mathbf{p}_\ell$ .

Based on the information collected in the database, the objective is to define a set of functions  $\psi_d(\cdot) : \mathbb{R}^{N_a} \mapsto \mathbb{R}$ , that associate to each entry  $\rho_\ell$  the corresponding output  $p_{\ell,d}$ , where  $d \in \{1, \dots, D\}$ ,  $D$  being the space's dimension and  $p_{\ell,d}$  is an element of  $\mathbf{p}_\ell = (p_{\ell,1}, \dots, p_{\ell,D})^\top$ . As we will show in Section III, kernel methods provide an elegant framework to determine these functions. Once  $\psi_d(\cdot)$  has been defined, a node's position, at a given time  $t$ , can be directly estimated in the *online* phase. Indeed, consider a node  $j$  to be localized at a given time  $t$ . This node receives signals from the  $N_a$  anchors in the environment, measures their RSSI values, and stores them in a vector  $\rho_j(t)$ . The estimated coordinates of the node  $j$  are then given by  $\hat{x}_{j,d}(t) = \psi_d(\rho_j(t))$ , where  $d \in \{1, \dots, D\}$ .

## III. DEFINING $\psi(\cdot)$ USING KERNEL METHODS

Let us consider a reproducing kernel  $\kappa : \mathbb{R}^{N_a} \times \mathbb{R}^{N_a} \mapsto \mathbb{R}$ , and let us denote by  $\mathcal{H}$  its reproducing kernel Hilbert space (RKHS) with inner product  $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ . From the reproducing property, we can say that every  $\psi_d(\cdot)$  of  $\mathcal{H}$  can be evaluated at any  $\rho_\ell \in \mathbb{R}^{N_a}$  by  $\psi_d(\rho_\ell) = \langle \psi_d(\cdot), \kappa(\cdot, \rho_\ell) \rangle_{\mathcal{H}}$ . The function  $\psi_d(\cdot)$  must minimize the mean quadratic error between the model's outputs  $\psi_d(\rho_\ell)$  and the desired outputs  $p_{\ell,d}$ :

$$\sum_{\ell=1}^N ((p_{\ell,d} - \psi_d(\rho_\ell))^2 + \eta \|\psi_d\|_{\mathcal{H}}^2), \quad (2)$$

where  $\eta$  is a regularization parameter that controls the tradeoff between the training error and the complexity of the solution, as measured by the norm in the RKHS. To minimize the

functional in (2), we take the functional derivative with respect to  $\psi_d(\cdot)$  and set it to zero [22], namely

$$-2 \sum_{\ell=1}^N (p_{\ell,d} - \langle \psi_d(\cdot), \kappa(\cdot, \rho_\ell) \rangle_{\mathcal{H}}) \kappa(\cdot, \rho_\ell) + 2\eta \psi_d(\cdot) = 0.$$

This optimality condition leads to the expression

$$\psi_d(\cdot) = \sum_{\ell=1}^N \frac{(p_{\ell,d} - \langle \psi_d(\cdot), \kappa(\cdot, \rho_\ell) \rangle_{\mathcal{H}})}{\eta} \kappa(\cdot, \rho_\ell). \quad (3)$$

Having  $\psi_d(\rho_\ell) = \langle \psi_d(\cdot), \kappa(\cdot, \rho_\ell) \rangle_{\mathcal{H}}$  from the reproducing property, we can write (3) as follows:

$$\psi_d(\cdot) = \sum_{\ell=1}^N \frac{(p_{\ell,d} - \psi_d(\rho_\ell))}{\eta} \kappa(\cdot, \rho_\ell)$$

Finally, we get the following model :

$$\psi_d(\cdot) = \sum_{\ell=1}^N \alpha_{\ell,d} \kappa(\rho_\ell, \cdot), \quad (4)$$

where

$$\alpha_{\ell,d} = \frac{p_{\ell,d} - \psi_d(\rho_\ell)}{\eta}. \quad (5)$$

By injecting expression (4) in the cost function (2), we get a least squares optimization problem that can be written in matrix form as follows :

$$(\mathbf{K} + \eta \mathbf{I}_N) \boldsymbol{\alpha}_d = \mathbf{P}_d, \quad (6)$$

where  $\boldsymbol{\alpha}_d = (\alpha_{1,d}, \dots, \alpha_{N,d})^\top$ ,  $\mathbf{P}_d = (p_{1,d}, \dots, p_{N,d})^\top$ ,  $\mathbf{I}_N$  is the  $N$ -by- $N$  identity matrix,  $\mathbf{K}$  is a  $N$ -by- $N$  matrix whose  $(i, j)$ -th entry is given by  $\kappa(\rho_i, \rho_j)$ .

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

$$\kappa(\rho_i, \rho_j) = \exp\left(\frac{-\|\rho_i - \rho_j\|^2}{2\sigma^2}\right),$$

where  $\sigma$  is the bandwidth of the Gaussian kernel. In this case, equation (4) approximates the unknown function by a weighted superposition of Gaussian functions. The weight  $\alpha_{\ell,d}$  of each Gaussian is taken such as to minimize the error on the training set. The bandwidth of the Gaussian kernel, together with the regularization parameter, control the degree of smoothness, noise tolerance, and generalization of the solution.

## IV. SIMULATION AND RESULTS

In order to evaluate the performance of our method, we tested it using various scenarios considering a  $2D$  environment. In the first part of this section, results are shown in the case of Matlab-generated data where RSSI values are obtained using the Okumura-Hata model [23]. However, experimental studies have shown that most of the state of the art localization algorithms, once deployed in real testbeds, achieve much worse performance than what was expected. Therefore, it is essential to test localization algorithms in the presence of real data. In the second part of this section, the proposed method is tested with real data gathered in a  $10m \times 10m$  real indoor environment [24]. In the third section, the results using the proposed method are compared to ones obtained with the well-known K-Nearest Neighbor localization algorithm, with different weighting functions.

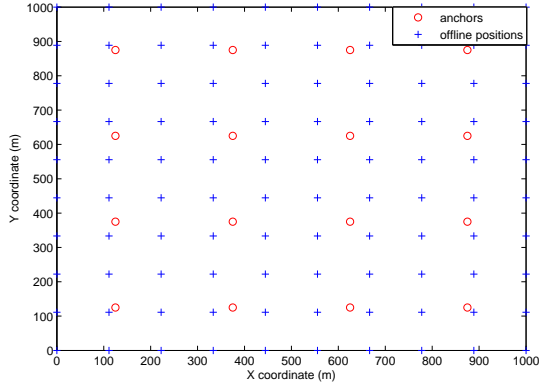


Fig. 1. Generated topology

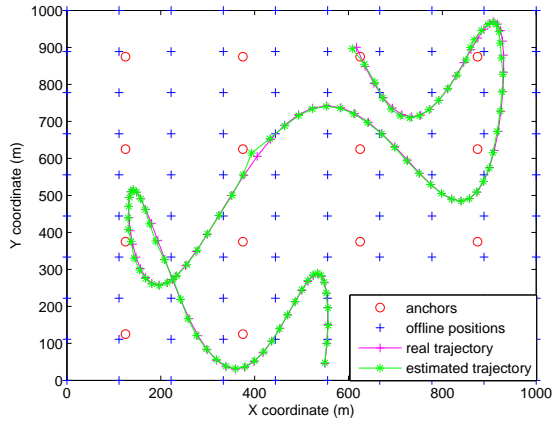


Fig. 2. Estimation of the trajectory in the absence of noise

#### A. Evaluation of the proposed method on simulated data

For the training phase, 100 *offline* positions and 16 anchors are generated in a uniformly distributed manner over a  $1000m \times 1000m$  area (Fig. 1). Anchors are assumed to be static. The RSSI values needed to construct the database of  $N$  couples  $(\rho_\ell, \mathbf{p}_\ell)$ , where  $\ell \in \{1, \dots, N\}$  ( $N = 100$ ), are generated using the Okumura-Hata model given by (1) with parameters set to  $n_P = 4$  and  $\rho_0 = 150dBm$ .

We generated a trajectory and calculated the RSSI values received by the moving node using the same RSSI model without noise; we then estimated the position of the node during its movement using our localization algorithm. Fig. 2 shows the generated trajectory and the estimated one. The two trajectories are superimposed. Let the estimation error be the root mean squared distances between the exact positions and the estimated ones. Then the estimation error is equal to  $1.063m$  for  $\eta = 2^{-20}$  and  $\sigma = 2^7$ . Results for different values of  $\eta$  and  $\sigma$  are shown in Table I. In Fig. 3, a zero-mean Gaussian white noise of standard deviation  $\sigma_b = 1dB$  is added to the RSSI values. The estimated trajectory follows the actual one and the estimation error is equal to  $10.656m$  for  $\eta = 2^{-12}$  and  $\sigma = 2^6$ ; results for different parameter values are shown in Table II.

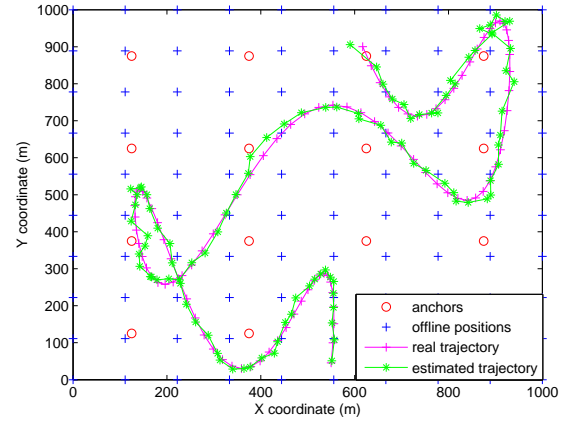


Fig. 3. Estimation of the trajectory in the presence of noise

TABLE I. ESTIMATION ERROR (IN METERS) FOR DIFFERENT VALUES OF  $\eta$  AND  $\sigma$  - MATLAB-SIMULATED DATA WITHOUT NOISE

$\sigma$	$\eta = 2^{-20}$	$\eta = 2^{-18}$	$\eta = 2^{-16}$	$\eta = 2^{-14}$	$\eta = 2^{-12}$
$2^4$	32.764	32.764	32.763	32.757	32.735
$2^5$	6.352	6.353	6.371	6.505	6.875
$2^6$	3.172	2.974	2.670	2.487	4.119
$2^7$	1.063	2.455	5.235	7.066	7.953
$2^8$	6.846	7.442	7.693	7.930	10.961
$2^9$	7.666	7.806	10.671	25.648	41.617
$2^{10}$	10.620	25.664	41.902	49.256	51.238

#### B. Evaluation of the proposed method using real data

The set of collected measurements used in this section are available from [24]. The authors deployed 48 EyesIFX nodes in a room measuring approximately  $10m \times 10m$ . The nodes are uniformly distributed over the area and suspended at approximately 75 centimeters from the ceiling. Furniture and people present in the room cause multi-path interferences affecting the collected RSSI values. Fig. 4 shows the topology of the testbed. We chose 5 nodes to be anchors in our simulation, leaving us with 43 nodes to use as *offline* positions. However, to get better results, we used the present data to generate 57 new *offline* positions in order to get a total of 100 *offline* positions (Fig. 5). To evaluate our algorithm in the case of a moving node, we also generated a trajectory using the data. Fig. 6 shows the generated trajectory and the estimated one. The estimation error is equal to  $0.195m$  for  $\eta = 2^{-19}$  and  $\sigma = 16$ . Fig. 7 shows how  $\eta$  and  $\sigma$  affect the estimation error, where  $\eta = 2^s$  with  $s \in \{-20, -19, \dots, -1\}$  and  $\sigma = 2^t$  with  $t \in \{1, 2, \dots, 10\}$ .

TABLE II. ESTIMATION ERROR (IN METERS) FOR DIFFERENT VALUES OF  $\eta$  AND  $\sigma$  - MATLAB-SIMULATED DATA WITH NOISE ( $\sigma_b = 1dB$ )

$\sigma$	$\eta = 2^{-20}$	$\eta = 2^{-18}$	$\eta = 2^{-16}$	$\eta = 2^{-14}$	$\eta = 2^{-12}$
$2^4$	35.347	36.113	35.766	35.625	35.600
$2^5$	12.628	12.795	12.787	13.032	12.514
$2^6$	12.991	11.545	12.266	11.377	10.656
$2^7$	16.547	14.533	13.165	13.201	12.349
$2^8$	15.071	13.653	12.405	12.816	14.033
$2^9$	12.817	12.771	14.457	26.987	43.108
$2^{10}$	14.818	27.711	42.690	49.467	52.322

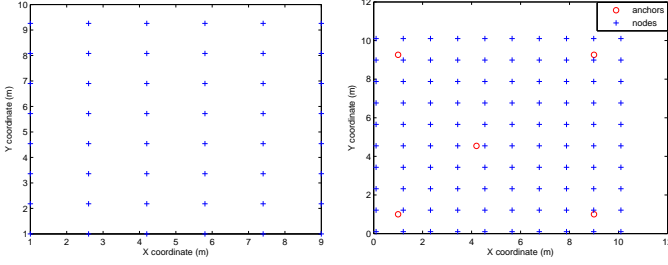


Fig. 4. Original testbed

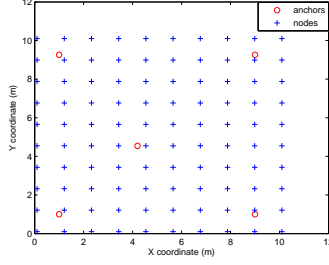


Fig. 5. Modified testbed

### C. Comparison to Weighted K-Nearest Neighbor algorithm

The Weighted K-Nearest Neighbor (WKNN) algorithm is a well-known localization algorithm based on the fingerprinting technique and the Euclidean distances between the RSSI values received by the node to be localized and the fingerprints in the database. Let us denote by  $D_{j,\ell} = \|\rho_j(t) - \rho_\ell\|$  the Euclidean distance between the RSSI vector  $\rho_j(t)$  of the node  $j$  we want to localize at time  $t$  and  $\rho_\ell$  of the database where  $\ell \in \{1, \dots, n\}$ . Let  $I_j$  be the set of indices of  $\rho_\ell$  yielding the  $K$  smallest distances  $D_{j,\ell}$  at time  $t$ .  $I_j$  of cardinality  $K$  corresponds to the set of the  $K$  nearest neighbors (*offline* positions) to the node  $j$  at time  $t$ . Then node  $j$ 's position is estimated by:

$$\hat{x}_j(t) = \sum_{k \in I_j} w_k p_k,$$

where  $p_k$  is one of the nearest *offline* positions neighboring the node, and  $w_k$  is the normalized weight factor depending on the  $k^{\text{th}}$  minimal distance. The weight factor is important in contributing to position accuracy; in fact, neighbors with high weight contribute more to position coordinates than elements with low weight, since weights and distances vary in inverse proportion to one another. Weight values used in simulations are listed in Table III.

Table IV shows the estimation error obtained while estimating the same trajectory of Fig. 2 (without adding noise to the RSSI values) using WKNN algorithm for  $K \in \{1, \dots, 15\}$  and for five different weights (Table III). The lowest error obtained using WKNN in this case is equal to  $16.863m$  when

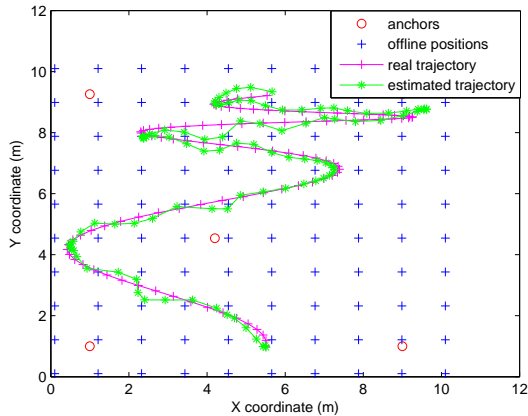


Fig. 6. Estimation of the trajectory (real data)

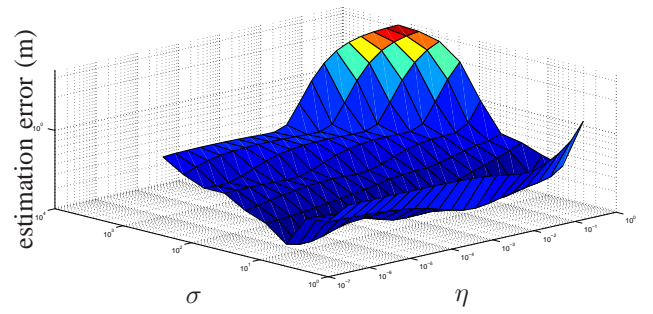


Fig. 7. Variations of the estimation error in terms of  $\eta$  and  $\sigma$

TABLE III. LIST OF SELECTED WEIGHT FACTORS  $w_k$

Type	Expression of $w_k$
"A"	$1/K$
"B"	$\frac{1/D_{j,k}}{\sum_{v \in I_j} 1/D_{j,v}}$
"C"	$\frac{1/D_{j,k}^2}{\sum_{v \in I_j} 1/D_{j,v}^2}$
"D"	$\frac{1/D_{j,k}^3}{\sum_{v \in I_j} 1/D_{j,v}^3}$
"E"	$\frac{\exp(-D_{j,k})}{\sum_{v \in I_j} \exp(-D_{j,v})}$

using weight "C" and  $K = 4$ . Using our proposed method, we found an estimation error of  $1.063m$  for  $\eta = 2^{-20}$  and  $\sigma = 2^7$  (see Table I).

After adding a zero-mean Gaussian white noise of standard deviation  $\sigma_b = 1dB$  to the RSSI values, we get the results in Table V, where the estimation error is presented in the case of WKNN algorithm. The lowest error obtained using WKNN is equal to  $17.644m$  when using weight "C" and  $K = 5$ . Our proposed method gives an estimation error of  $10.656m$  for  $\eta = 2^{-12}$  and  $\sigma = 6$  (see Table II) which is also smaller than the best result obtained with WKNN.

We now consider the same scenario as in Section IV.B (real RSSI values are used). Table VI shows the estimation error obtained while estimating the trajectory of Fig. 6 using WKNN algorithm. The lowest error obtained is equal to  $0.340m$  when using weight "E" and  $K \in \{10, 11, 12, 13\}$ . Our proposed method gives an estimation error of  $0.195m$  for  $\eta = 2^{-19}$  and  $\sigma = 16$  which is a better result than the one obtained with WKNN.

## V. CONCLUSION

In this paper, we presented a new localization algorithm based on radio-location fingerprinting and kernel methods. The simulation results show that the proposed method allows a more accurate localization than the WKNN algorithm, for different weight models and parameters. We also analyzed the behavior of our algorithm in the presence of real data, and the results were satisfying. However, we cannot but notice that the results obtained in the case of Matlab simulated data without the additive noise are more than excellent, leaving us with the conclusion that, in the case of an accurate radio channel model, the proposed model technique overcome many existing localization techniques in terms of accuracy.



TABLE IV. ESTIMATION ERROR (IN METERS) BETWEEN REAL AND ESTIMATED TRAJECTORY - MATLAB-SIMULATED DATA WITHOUT NOISE

K	"A"	"B"	"C"	"D"	"E"
1	48.376	48.376	48.376	48.376	48.376
2	36.538	27.603	27.647	29.801	38.935
3	28.882	20.535	20.108	23.030	37.641
4	29.114	19.865	16.863	19.516	37.245
5	28.969	20.062	16.966	19.242	37.230
6	35.913	24.720	18.424	18.959	37.228
7	35.705	25.655	19.472	19.592	37.227
8	38.163	27.503	20.287	19.618	37.227
9	39.979	29.327	21.120	19.636	37.227
10	37.017	27.591	19.970	18.965	37.227
11	36.504	27.353	19.829	18.728	37.227
12	36.870	27.059	19.468	18.456	37.227
13	38.415	27.958	19.664	18.337	37.227
14	39.871	28.105	19.491	18.087	37.227
15	39.939	28.295	19.825	18.304	37.227

TABLE V. ESTIMATION ERROR (IN METERS) BETWEEN REAL AND ESTIMATED TRAJECTORY - MATLAB-SIMULATED DATA WITH NOISE ( $\sigma_b = 1dB$ )

K	"A"	"B"	"C"	"D"	"E"
1	49.205	49.205	49.205	49.205	49.205
2	36.639	28.390	27.536	29.491	40.328
3	30.285	22.605	21.218	23.506	39.121
4	27.907	20.414	17.805	20.167	38.768
5	28.792	20.814	17.644	19.474	38.698
6	36.218	26.035	19.792	19.737	38.691
7	36.134	26.708	20.363	19.878	38.690
8	38.966	28.550	20.931	19.734	38.690
9	40.597	30.016	21.945	20.046	38.690
10	37.429	28.559	20.986	19.413	38.690
11	37.629	28.677	20.838	19.126	38.690
12	37.398	27.831	20.041	18.618	38.690
13	39.135	28.912	20.384	18.541	38.690
14	40.254	29.231	20.512	18.443	38.690
15	40.334	29.079	20.599	18.471	38.690

#### ACKNOWLEDGMENT

The authors would like to thank the Champagne-Ardenne Region in France for funding this project.

#### 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] K. Sohrawy, D. Minoli, and T. Znati, *Wireless Sensor Networks: Technology, Protocols, and Applications*. John Wiley, 2007.
- [3] "Special issue on self-organizing distributed collaborative sensor networks," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 4, 2005.
- [4] S. H. Lee, S. Lee, H. Song, and H. S. Lee, "Wireless sensor network design for tactical military applications : Remote large-scale environments," in *Military Communications Conference, 2009. MILCOM 2009. IEEE*, oct. 2009, pp. 1 – 7.
- [5] L. Yu, N. Wang, and X. Meng, "Real-time forest fire detection with wireless sensor networks," in *Wireless Communications, Networking and Mobile Computing, 2005. Proceedings. 2005 International Conference on*, vol. 2, sept. 2005, pp. 1214 – 1217.
- [6] G. F. S. De Vito, "Wireless chemical sensor networks for air quality monitoring," in *14th International Meeting on Chemical Sensors - IMCS 2012*, may 2012, pp. 641 – 644.
- [7] B. Hofmann-Wellenhof, H. Lichtenegger, and J. Collins, *Global positioning system: Theory and practice*. Springer-Verlag, 1994.
- [8] J. Shen and A. Molisch, "Passive location using toa measurements," *IEEE International Conference on Ultra-Wideband (ICUWB)*, pp. 253 – 257, 2011.
- [9] R. Eickhoff, F. Ellinger, R. Mosshammer, R. Weigel, A. Ziroff, and M. Huemer, "3d-accuracy improvements for toa based wireless local positioning systems," in *GLOBECOM Workshops, 2008 IEEE*, 2008.

TABLE VI. ESTIMATION ERROR (IN METERS) BETWEEN REAL AND ESTIMATED TRAJECTORY - REAL DATA (SECTION IV.B)

K	"A"	"B"	"C"	"D"	"E"
1	0.500	0.500	0.500	0.500	0.500
2	0.543	0.486	0.475	0.475	0.467
3	0.480	0.431	0.426	0.433	0.406
4	0.550	0.432	0.410	0.417	0.400
5	0.564	0.438	0.404	0.408	0.383
6	0.606	0.453	0.405	0.408	0.378
7	0.596	0.448	0.391	0.400	0.362
8	0.618	0.460	0.382	0.393	0.348
9	0.644	0.480	0.384	0.389	0.341
10	0.636	0.485	0.388	0.389	0.340
11	0.615	0.483	0.394	0.393	0.340
12	0.620	0.483	0.394	0.393	0.340
13	0.649	0.496	0.398	0.396	0.340
14	0.714	0.527	0.407	0.397	0.341
15	0.758	0.547	0.413	0.399	0.341

- [10] P. Rong and M. Sichitiu, "Angle of arrival localization for wireless sensor networks," in *Sensor and Ad Hoc Communications and Networks, 2006. SECON '06. 2006 3rd Annual IEEE Communications Society on*, vol. 1, sept. 2006, pp. 374 – 382.
- [11] 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.
- [12] G. Zanca, F. Zorzi, A. Zanella, and M. Zorzi, "Experimental comparison of rssi-based localization algorithms for indoor wireless sensor networks," in *Proceedings of the workshop on Real-world wireless sensor networks*, ser. REALWSN '08, 2008, pp. 1–5.
- [13] F. Mourad, H. Snoussi, F. Abdallah, and C. Richard, "Model-free interval-based localization in manetes," *IEEE 13th Digital Signal Processing Workshop and IEEE 5th Signal Processing Education Workshop (DSP/SPE)*, pp. 474–479, 2009.
- [14] —, "A robust localization algorithm for mobile sensors using belief functions and interval theory," *IEEE Transactions on Vehicular Technology*, vol. 60, no. 4, pp. 1799 – 1811, 2011.
- [15] 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.
- [16] P. Honeine and C. Richard, "Preimage problem in kernel-based machine learning," *IEEE Signal Processing Magazine*, vol. 28, no. 2, pp. 77–88, 2011.
- [17] 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, 2005 International Conference on*, vol. 2, june 2005, pp. 1569 – 1574 vol.2.
- [18] J. Robles, M. Deicke, and R. Lehnert, "3d fingerprint-based localization for wireless sensor networks," in *Positioning Navigation and Communication (WPNC), 2010 7th Workshop on*, march 2010, pp. 77 – 85.
- [19] W. N. W. Meng, W. Xiao and L. Xie, "Secure and robust wi-fi fingerprinting indoor localization," in *IEEE International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, Guimaraes, Portugal, 31-23 Sept. 2011, pp. 1–7.
- [20] 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.
- [21] N. Aronszajn, "Theory of reproducing kernels," *Trans. Amer. Math. Soc.*, vol. 68, no. 3, pp. 337 – 404, 1950.
- [22] T. Poggio and S. Smale, "The mathematics of learning: Dealing with data," in *Neural Networks and Brain, 2005. ICNN B '05. International Conference on*, vol. 1, oct. 2005, pp. PL–5 – PL–23.
- [23] A. Medeis and A. Kajackas, "On the use of the universal okumura-hata propagation prediction model in rural areas," *Vehicular Conference Proceedings*, vol. 3, 2000.
- [24] G. Zanca, F. Zorzi, and A. Zanella, "Rssi measurements in indoor wireless sensor networks." [Online]. Available: <http://telecom.dei.unipd.it/pages/read/59/>